

**PENGEMBANGAN APLIKASI *MOBILE* PELAPORAN KELUHAN  
PELANGGAN PDAM MENGGUNAKAN FITUR *GEOTAGGING*  
BERBASIS ANDROID**

**(Studi Kasus: PDAM Tirta Tuah Benua Kutai Timur)**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Anggi Mahadika Purnomo

NIM: 145150200111052



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## PENGESAHAN

PENGEMBANGAN APLIKASI *MOBILE* PELAPORAN KELUHAN PELANGGAN PDAM  
MENGUNAKAN FITUR *GEOTAGGING* BERBASIS ANDROID  
(Studi Kasus: PDAM Tirta Tuah Benua Kutai Timur)


### SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Anggi Mahadika Purnomo  
NIM: 145150200111052

Skrripsi ini telah diuji dan dinyatakan lulus pada  
27 Desember 2018  
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I


  
Bayu Priyambadha, S.Kom, M.Kom  
NIP: 19820909 200812 1 004

Dosen Pembimbing II

  
Agi Putra Kharisma, S.T, M.T  
NIK: 201304 860430 1 001

Mengetahui  
Ketua Jurusan Teknik Informatika



  
Tri Astoto Kurniawan, S.T, M.T, Ph.D  
NIP: 19710518 200312 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 12 Desember 2018



Anggi Mahadika Purnomo

NIM: 145150200111052



## KATA PENGANTAR

Segala puji bagi Allah SWT yang telah melimpahkan rahmat serta hidayah-Nya sehingga penulis dapat menyelesaikan penulisan skripsi dengan judul “Pengembangan Aplikasi *Mobile* Pelaporan Keluhan Pelanggan PDAM menggunakan Fitur *Geotagging* Berbasis Android (Studi Kasus: PDAM Tirta Tuah Benua Kutai Timur)”.

Pada kesempatan ini penulis juga mengucapkan terimakasih kepada seluruh pihak yang telah membantu penulis selama pengerjaan skripsi ini, diantaranya:

1. Allah SWT yang memberikan kesehatan serta kemudahan dalam segala proses pengerjaan skripsi.
2. Kedua orang tua penulis, yaitu Abah Rochmad Edy Purnomo dan Ibu Binti Usdalifah yang tak pernah henti dalam memanjatkan doa serta memberikan dukungan dan motivasi kepada penulis untuk segera menyelesaikan skripsi ini.
3. Bapak Bayu Priyambadha S.Kom, M.Kom selaku dosen pembimbing I yang telah meluangkan waktu untuk selalu sabar membimbing serta memberikan saran dan ilmu yang bermanfaat dalam pengerjaan skripsi ini.
4. Bapak Agi Putra Kharisma S.T, M.T selaku dosen pembimbing II yang juga telah meluangkan waktu untuk membimbing serta memberikan saran dan ilmu yang bermanfaat dalam pengerjaan skripsi ini.
5. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Suprpto, S.T, M.T, dan Bapak Edy Santoso, S.Si, M.Kom selaku Dekan, Wakil Dekan 1, Wakil Dekan 2 dan Wakil Dekan 3 Fakultas Ilmu Komputer, Universitas Brawijaya.
6. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D dan Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Jurusan Teknik Informatika dan Kepala Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Brawijaya.
7. Seluruh Dosen Fakultas Ilmu Komputer, Universitas Brawijaya yang telah memberikan ilmu yang bermanfaat bagi penulis.
8. Karyawan PDAM Tirta Tuah Benua yang bersedia menjadi narasumber serta responden untuk penelitian ini.
9. Sahabat serta kerabat penulis yang selalu memberikan semangat dan menjadi pendengar keluh kesah penulis selama mengerjakan skripsi ini dari awal hingga selesai.

10. Teman-teman Informatika 2014 yang telah berjuang bersama.

Penulis sadar masih banyak kekurangan dalam penyusunan skripsi ini baik dalam penyajian materi maupun pembahasan. Demi menyempurnakan penelitian ini, penulis sangat mengharapkan adanya kritik serta saran yang membangun. Semoga karya tulis ini dapat bermanfaat bagi pihak-pihak yang membutuhkan.

Malang, 12 Desember 2018

Penulis

anggimahadika@gmail.com





## ABSTRAK

PDAM Tirta Tuah Benua Kutai Timur sebagai suatu badan pengelola air bersih di Sangatta Utara berupaya untuk memaksimalkan pelayanan dan menangani pengaduan keluhan dengan cepat dan tepat guna merealisasikan misi yang ada. Berdasarkan wawancara dengan karyawan PDAM, dapat disimpulkan bahwa prosedur pelaporan yang manual dinilai kurang efektif karena kurangnya informasi mengenai bagian yang menangani laporan pelanggan sehingga pelanggan akan kesulitan ketika akan melapor melalui telepon dan PDAM tidak dapat memberikan umpan balik kepada pelanggan. Selain itu pelanggan juga harus datang langsung untuk menyerahkan bukti tagihan air untuk laporan tagihan yang tidak sesuai pemakaian. Masalah lain yaitu mengenai lokasi pelanggan yang diberikan secara manual akan mempersulit pihak PDAM dalam mencari alamat apabila lokasi berada di pelosok. Untuk dapat menangani permasalahan diatas, solusi yang penulis tawarkan yaitu dengan mengembangkan aplikasi *mobile* pelaporan keluhan pelanggan PDAM dimana pelanggan dapat melaporkan keluhannya melalui aplikasi *mobile* dan karyawan dapat memberikan umpan balik melalui website admin. Penulis menggunakan model waterfall dalam mengembangkan aplikasi. Pada penelitian ini menggunakan fitur *geotagging* untuk menambahkan lokasi pada gambar yang dilampirkan oleh pelanggan. Kemudian dilakukan pengujian unit, validasi, serta *compatibility*. Berdasarkan pengujian validasi didapatkan hasil bahwa semua fungsi bernilai valid yang berarti telah sesuai dengan kebutuhan yang telah didefinisikan. Hasil dari pengujian *compatibility* menunjukkan bahwa aplikasi *mobile* dapat digunakan pada 4 versi OS Android yang berbeda.

**Kata kunci:** aplikasi pelaporan, *geotagging*, pengembangan aplikasi.

## ABSTRACT

*PDAM Tirta Tuah Benua East Kutai as regional owned enterprises that manages clean water in North Sangatta seeks to maximizing service and handling customer complaints quickly and precisely to realize the PDAM's mission. Based on interviews with PDAM, it can be concluded that the existing manually reporting procedure is considered ineffective because of lack of information about the part that handles customer reports so customers will have difficulties when reporting their complaints via telephone and the PDAM can't give a feedback to the customer. In addition, customers are also required to come directly to submit the payment invoices for bill complaints that are not according to usage. Another problem is about the location of the customers that given manually wich will make PDAM's employee feels difficult to find the address if the location is in a remote area. To handle the problems above, the solution that the author offers is to develop a mobile customer complaint reporting application where customers can report their complaints through a mobile application and employees can give a feedback through the admin's website. The author used waterfall model to developing the software. In this research using the geotagging feature to add location to the images sent by the customers. Then do the unit, validation, and compatibility testing. Based on the validation testing, it shows that all functions are valid, which means they are in accordance with the the defined requirements. And the result of compatibility testing show that mobile applications can be used on 4 different Android OS versions.*

**Keywords:** application development, geotagging, reporting application, software engineering.

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT .....	vii
DAFTAR ISI .....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah .....	3
1.3 Tujuan Penelitian .....	3
1.4 Manfaat.....	3
1.5 Batasan masalah .....	4
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN .....	6
2.1 Kajian Pustaka .....	6
2.2 Pelaporan Keluhan Pelanggan .....	8
2.3 Rekayasa Perangkat Lunak .....	9
2.4 <i>Software Development Life Cycle</i> .....	10
2.4.1 <i>Waterfall Model</i> .....	11
2.5 <i>Unified Modeling Language</i> .....	11
2.5.1 <i>Use Case Diagram</i> .....	12
2.5.2 <i>Sequence Diagram</i> .....	14
2.5.3 <i>Class Diagram</i> .....	15
2.6 Android .....	16
2.6.1 Versi Android.....	16
2.7 <i>Website</i> .....	17
2.8 GPS .....	18
2.9 Google Map API .....	19
2.10 <i>Geotagging</i> .....	19



2.11 Pengujian Perangkat Lunak .....	19
2.11.1 Pengujian <i>Whitebox</i> .....	20
2.11.2 Pengujian <i>Blackbox</i> .....	20
2.11.3 Pengujian Validasi .....	20
2.11.4 Pengujian <i>Compatibility</i> .....	21
BAB 3 METODOLOGI .....	22
3.1 Studi Literatur .....	23
3.2 Pengumpulan Data .....	24
3.3 Analisis Kebutuhan .....	24
3.4 Perancangan Sistem.....	24
3.5 Implementasi .....	25
3.6 Pengujian dan Analisis .....	25
3.7 Kesimpulan dan Saran .....	26
BAB 4 ANALISIS KEBUTUHAN .....	27
4.1 Gambaran Umum Aplikasi .....	27
4.2 Identifikasi Aktor.....	27
4.3 Analisis Kebutuhan Fungsional .....	28
4.3.1 <i>Use Case</i> Diagram.....	30
4.3.2 <i>Use Case</i> Skenario .....	32
4.4 Analisis Kebutuhan Non Fungsional .....	39
BAB 5 PERANCANGAN DAN IMPLEMENTASI .....	40
5.1 Perancangan Perangkat Lunak .....	40
5.1.1 Perancangan Arsitektur Sistem.....	41
5.1.2 Perancangan Basis Data .....	41
5.1.3 Perancangan <i>Sequence</i> Diagram.....	44
5.1.4 Perancangan <i>Class</i> Diagram .....	47
5.1.5 Perancangan Algoritme.....	49
5.1.6 Perancangan Antarmuka.....	50
5.2 Implementasi .....	62
5.2.1 Spesifikasi Sistem .....	62
5.2.2 Batasan Implementasi.....	63
5.2.3 Implementasi Sistem.....	64

BAB 6 PENGUJIAN DAN ANALISIS.....	75
6.1 Pengujian Unit.....	75
6.1.1 Pengujian Unit operasi onClick() pada kelas UploadActivity.java	75
6.1.2 Pengujian Unit Operasi onCreateView() pada kelas CameraFragment.java .....	77
6.1.3 Pengujian Unit Operasi getAddress() pada Kelas GeotagImage.java .....	79
6.2 Pengujian Validasi .....	81
6.2.1 Pengujian Validasi Melakukan Registrasi .....	82
6.2.2 Pengujian Validasi Melakukan <i>Login User</i> .....	83
6.2.3 Pengujian Validasi Melihat <i>Home User</i> .....	84
6.2.4 Pengujian Validasi Melaporkan Keluhan.....	84
6.2.5 Pengujian Validasi Mengambil Gambar dari Kamera .....	85
6.2.6 Pengujian Validasi Mengambil Gambar dari <i>Gallery</i> .....	86
6.2.7 Pengujian Validasi Melihat Laporan Saya .....	86
6.2.8 Pengujian Validasi Melihat Detail Laporan oleh User.....	87
6.2.9 Pengujian Validasi Merubah Laporan .....	87
6.2.10 Pengujian Validasi Menghapus Laporan User.....	88
6.2.11 Pengujian Validasi Melakukan Logout User .....	88
6.2.12 Pengujian Validasi Melakukan <i>Login Admin</i> .....	88
6.2.13 Pengujian Validasi Melihat Daftar Laporan .....	89
6.2.14 Pengujian Validasi Melihat Detail Laporan oleh Admin.....	89
6.2.15 Pengujian Validasi Merubah Status Laporan .....	90
6.2.16 Pengujian Validasi Menghapus Laporan oleh Admin .....	90
6.2.17 Pengujian Validasi Melihat Daftar Pelanggan .....	90
6.2.18 Pengujian Menghapus Pelanggan .....	91
6.2.19 Pengujian Validasi Melakukan Logout Admin.....	91
6.3 Pengujian <i>Compatibility</i> .....	91
6.4 Analisis Hasil Pengujian.....	94
6.4.1 Pengujian Unit.....	95
6.4.2 Pengujian Validasi .....	95
6.4.3 Pengujian <i>Compatibility</i> .....	95
BAB 7 KESIMPULAN DAN SARAN .....	96

7.1 Kesimpulan.....	96
7.2 Saran .....	96
DAFTAR PUSTAKA.....	98



## DAFTAR TABEL

Tabel 2.1 Kajian Pustaka .....	7
Tabel 2.2 Simbol pada <i>Use Case Diagram</i> .....	12
Tabel 2.3 Simbol pada <i>Sequence Diagram</i> .....	14
Tabel 2.4 Versi Android .....	16
Tabel 4.1 Identifikasi Aktor .....	28
Tabel 4.2 Kebutuhan Fungsional Aplikasi Untuk Tamu.....	28
Tabel 4.3 Kebutuhan Fungsional Aplikasi Untuk User .....	28
Tabel 4.4 Kebutuhan Fungsional Admin .....	29
Tabel 4.5 <i>Use Case</i> Skenario Melakukan Registrasi .....	32
Tabel 4.6 <i>Use Case</i> Skenario Melakukan Login User .....	32
Tabel 4.7 <i>Use Case</i> Skenario Melakukan Login Admin .....	33
Tabel 4.8 <i>Use Case</i> Skenario Melihat <i>Home</i> User .....	33
Tabel 4.9 <i>Use Case</i> Skenario Melaporkan Keluhan oleh user.....	33
Tabel 4.10 <i>Use Case</i> Skenario Mengambil Gambar dari Kamera .....	34
Tabel 4.11 <i>Use Case</i> Skenario Mengambil Gambar dari <i>Gallery</i> .....	35
Tabel 4.12 <i>Use Case</i> Skenario Melihat Laporan Saya .....	35
Tabel 4.13 <i>Use Case</i> Skenario Melihat Detail Laporan oleh User.....	35
Tabel 4.14 <i>Use Case</i> Skenario Melihat Detail Laporan oleh Admin.....	36
Tabel 4.15 <i>Use Case</i> Skenario Merubah Laporan .....	36
Tabel 4.16 <i>Use Case</i> Skenario Menghapus Laporan User.....	36
Tabel 4.17 <i>Use Case</i> Skenario Melihat Daftar Laporan Admin .....	37
Tabel 4.18 <i>Use Case</i> Skenario Menghapus Laporan oleh Admin.....	37
Tabel 4.19 <i>Use Case</i> Skenario Merubah Status Laporan .....	37
Tabel 4.20 <i>Use Case</i> Skenario Melihat Daftar Pelanggan .....	38
Tabel 4.21 <i>Use Case</i> Skenario Menghapus Pelanggan.....	38
Tabel 4.22 <i>Use Case</i> Skenario Melakukan <i>Logout</i> User .....	38
Tabel 4.23 <i>Use Case</i> Skenario Melakukan <i>Logout</i> Admin.....	39
Tabel 4.24 Kebutuhan Non-Fungsional.....	39
Tabel 5.1 Rancangan Struktur Tabel <i>User</i> .....	42
Tabel 5.2 Rancangan Struktur Tabel Member .....	43
Tabel 5.3 Rancangan Struktur Tabel Complaint.....	43

Tabel 5.4 Spesifikasi Perangkat Keras Laptop .....	63
Tabel 5.5 Spesifikasi Perangkat Keras <i>Smartphone</i> .....	63
Tabel 5.6 Spesifikasi Perangkat Lunak .....	63
Tabel 5.7 <i>Syntax Create Tabel Member</i> .....	64
Tabel 5.8 <i>Syntax Create Tabel User</i> .....	64
Tabel 5.9 <i>Syntax Create Tabel Complaint</i> .....	65
Tabel 6.1 Hasil Pengujian Unit Operasi <i>onClick()</i> pada Kelas <i>UploadActivity.java</i> .....	77
Tabel 6.2 Hasil Pengujian Unit Operasi <i>onCreateView()</i> pada Kelas <i>CameraFragment.java</i> .....	78
Tabel 6.3 Hasil Pengujian Unit Operasi <i>getAddress()</i> pada Kelas <i>GeotagImage.java</i> .....	81
Tabel 6.4 Kasus Uji Berhasil Melakukan Registrasi .....	82
Tabel 6.5 Kasus Uji Gagal Melakukan Registrasi dengan Tidak Mengisi Field <i>Username</i> .....	82
Tabel 6.6 Kasus Uji Berhasil Melakukan <i>Login User</i> .....	83
Tabel 6.7 Kasus Uji Gagal Melakukan <i>Login User</i> dengan Menginputkan <i>Password</i> yang Salah .....	83
Tabel 6.8 Kasus Uji Berhasil Melihat <i>Home User</i> .....	84
Tabel 6.9 Kasus Uji Berhasil Melaporkan Keluhan .....	84
Tabel 6.10 Kasus Uji Gagal Melaporkan Keluhan dengan Tidak Mengisi <i>Field Deskripsi</i> .....	84
Tabel 6.11 Kasus Uji Berhasil Mengambil Gambar dari Kamera .....	85
Tabel 6.12 Kasus Uji Gagal Mengambil Gambar dari Kamera dengan Tidak Menyalakan GPS <i>Smartphone</i> .....	85
Tabel 6.13 Kasus Uji Berhasil Mengambil Gambar dari <i>Gallery</i> .....	86
Tabel 6.14 Kasus Uji Gagal Mengambil Gambar dari <i>Gallery</i> dengan Mengambil Gambar yang Tidak Memiliki Informasi Lokasi .....	86
Tabel 6.15 Kasus Uji Berhasil Melihat laporan Saya .....	86
Tabel 6.16 Kasus Uji Berhasil Melihat Detail Laporan oleh User .....	87
Tabel 6.17 Kasus Uji Berhasil Merubah Laporan .....	87
Tabel 6.18 Kasus Uji Gagal Merubah Laporan dengan Tidak Mengisi <i>Field Deskripsi</i> .....	87
Tabel 6.19 Kasus Uji Berhasil Menghapus Laporan .....	88
Tabel 6.20 Kasus Uji Berhasil melakukan Logout .....	88
Tabel 6.21 Kasus Uji Berhasil Melakukan <i>Login Admin</i> .....	88



Tabel 6.22 Kasus Uji Gagal Melakukan <i>Login</i> Admin dengan Mengisi <i>Username</i> yang Salah .....	89
Tabel 6.23 Kasus Uji Berhasil Melihat Daftar Laporan.....	89
Tabel 6.24 Kasus Uji Berhasil Melihat Detail Laporan oleh Admin .....	89
Tabel 6.25 Kasus Uji Berhasil Merubah Status Laporan .....	90
Tabel 6.26 Kasus Uji Berhasil Menghapus Laporan oleh Admin.....	90
Tabel 6.27 Kasus Uji Berhasil Melihat Daftar Pelanggan .....	90
Tabel 6.28 Kasus Uji Berhasil Menghapus Pelanggan.....	91
Tabel 6.29 Kasus Uji Berhasil Melakukan Logout Admin .....	91
Tabel 6.30 Hasil Pengujian <i>Compatibility</i> .....	94



## DAFTAR GAMBAR

Gambar 2.1 Diagram Alur Status Penanganan Keluhan Pelanggan.....	9
Gambar 2.2 Tahapan Umum Rekayasa Perangkat Lunak .....	10
Gambar 2.3 Diagram Alir <i>Waterfall</i> Model .....	11
Gambar 2.4 Contoh <i>Class</i> Diagram .....	15
Gambar 2.5 Trilaterasi dalam GPS .....	18
Gambar 3.1 Diagram Alir Metodologi Penelitian .....	22
Gambar 4.1 Tahapan Analisis Kebutuhan .....	27
Gambar 4.2 Use Case Diagram User .....	31
Gambar 4.3 Use Case Diagram <i>Website</i> Admin .....	31
Gambar 5.1 Tahapan Perancangan Perangkat Lunak .....	40
Gambar 5.2 Arsitektur Sistem dari Aplikasi Pelaporan Keluhan Pelanggan PDAM .....	41
Gambar 5.3 <i>Entity Relationship Diagram</i> .....	42
Gambar 5.4 <i>Sequence</i> Diagram Melaporkan Keluhan .....	44
Gambar 5.5 <i>Sequence</i> Diagram Mengambil Gambar dari Kamera .....	45
Gambar 5.6 <i>Sequence</i> Diagram Mengambil Foto dari Gallery .....	46
Gambar 5.7 <i>Class</i> Diagram Aplikasi <i>Mobile</i> untuk <i>User</i> .....	47
Gambar 5.8 <i>Class</i> Diagram <i>Website</i> Admin .....	48
Gambar 5.9 <i>Sitemap</i> Aplikasi .....	51
Gambar 5.10 <i>Screenflow</i> Aplikasi .....	51
Gambar 5.11 <i>Sitemap</i> Website .....	52
Gambar 5.12 <i>Screenflow</i> <i>Website</i> Admin .....	53
Gambar 5.13 Antarmuka Halaman Login .....	54
Gambar 5.14 Antarmuka Halaman Registrasi .....	54
Gambar 5.15 Antarmuka Halaman Home .....	55
Gambar 5.16 Antarmuka Halaman <i>Camera</i> .....	56
Gambar 5.17 Antarmuka Halaman <i>Gallery</i> .....	56
Gambar 5.18 Antarmuka Halaman Laport .....	57
Gambar 5.19 Antarmuka Halaman Laporan Saya .....	57
Gambar 5.20 Antarmuka Halaman Detail Laporan .....	58
Gambar 5.21 Antarmuka Halaman Login Admin .....	58

Gambar 5.22 Antarmuka Halaman Daftar Laporan Admin.....	59
Gambar 5.23 Antarmuka Halaman Detail Laporan Admin .....	60
Gambar 5.24 Antarmuka Halaman Ubah Status.....	61
Gambar 5.25 Antarmuka Halaman Daftar Pelanggan.....	62
Gambar 5.26 <i>Physical Data Model</i> basis data.....	65
Gambar 5.27 Implementasi Antarmuka Halaman <i>Login User</i> .....	68
Gambar 5.28 Implementasi Antarmuka Halaman Registrasi.....	69
Gambar 5.29 Implementasi Antarmuka Halaman <i>Home User</i> .....	69
Gambar 5.30 Implementasi Antarmuka Halaman <i>Camera</i> .....	70
Gambar 5.31 Implementasi Antarmuka Halaman <i>Gallery</i> .....	70
Gambar 5.32 Implementasi Antarmuka Halaman Form Pelaporan Keluhan .....	71
Gambar 5.33 Implementasi Antarmuka Halaman Laporan Saya.....	71
Gambar 5.34 Implementasi Antarmuka Halaman Detail Laporan.....	72
Gambar 5.35 Implementasi Antarmuka Halaman <i>Login Admin</i> .....	72
Gambar 5.36 Implementasi Antarmuka Halaman Daftar Laporan.....	73
Gambar 5.37 Implementasi Antarmuka Halaman Detail Laporan Admin .....	73
Gambar 5.38 Implementasi Antarmuka Halaman Ubah Status.....	74
Gambar 5.39 Implementasi Antarmuka Halaman Daftar Pelanggan .....	74
Gambar 6.1 Tampilan Aplikasi pada Android Versi 5.1 Lollipop.....	92
Gambar 6.2 Tampilan Aplikasi pada Android Versi 6.0.1 Marshmallow .....	92
Gambar 6.3 Tampilan Aplikasi pada Android Versi 7.0 Nougat.....	93
Gambar 6.4 Tampilan Aplikasi pada Android Versi 8.0 Oreo .....	93

## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Air merupakan senyawa yang sangat penting dan tidak dapat dipisahkan dari keberlangsungan kehidupan sehari-hari. Tanpa adanya ketersediaan air bersih yang cukup, kelangsungan hidup manusia akan sangat terganggu. Di Indonesia, pengelolaan air bersih telah ditangani oleh suatu badan usaha milik daerah yang bernama Perusahaan Daerah Air Minum (PDAM). PDAM didirikan di setiap kota yang ada di Indonesia guna memenuhi kebutuhan masyarakat akan air bersih yang sangat tinggi, salah satunya yang ada di Kota Sangatta Utara. PDAM Tirta Buah Benua Kutai Timur merupakan PDAM yang berada di kota Sangatta Utara, Kutai Timur. Salah satu Misi dari PDAM Tirta Buah Benua Kutai Timur yaitu meningkatkan cakupan pelayanan di wilayah perkotaan hingga 80% dan di wilayah pedesaan hingga 60% (Tirta Buah Benua, 2017). Guna merealisasikan misi yang telah ada, PDAM Tirta Buah Benua Kutai Timur memperlebar cakupan pelayanannya serta meningkatkan kepuasan pelanggan dengan cara melayani pelanggan dengan maksimal dan menangani pengaduan keluhan pelanggan mengenai masalah air dengan cepat dan tepat.

Menurut salah satu karyawan bagian Hubungan Langsung pada PDAM Tirta Buah Benua Kutai Timur, macetnya aliran air akibat kurangnya stok air ataupun terdapat kebocoran pipa di titik tertentu merupakan permasalahan yang sering dihadapi saat ini. Berbagai permasalahan lain seperti kualitas air yang kurang bersih atau air berbau tidak sedap serta tingginya tagihan air yang tidak sesuai pemakaian juga sering dilaporkan melalui telfon oleh pelanggan. Selama ini pelanggan yang ingin melaporkan keluhannya bisa datang langsung ataupun menelfon PDAM pada bagian yang menangani permasalahan dari pelanggan tersebut, dimana nomor masing-masing bagian dapat dilihat di website resmi PDAM Tirta Buah Benua yaitu [www.pdam-kutaitimur.com](http://www.pdam-kutaitimur.com). Namun berdasarkan hasil observasi yang telah penulis lakukan, prosedur yang ada sekarang dinilai kurang efektif karena kurangnya informasi yang diberikan mengenai bagian-bagian yang menangani laporan pelanggan, sehingga pelanggan masih kebingungan ketika ingin menelfon untuk membuat laporan dan PDAM tidak bisa memberikan umpan balik terhadap laporan pelanggan yang ada apabila dilaporkan melalui telfon. Selain itu untuk permasalahan tagihan air yang dinilai tidak sesuai pemakaian, pelanggan wajib datang langsung untuk memberikan bukti fisik tagihan air pada bulan tersebut sehingga cukup merepotkan pelanggan. Masalah lain yang timbul dan tidak kalah penting yaitu mengenai lokasi pelanggan tersebut berada. Dimana selama ini pelanggan memberikan informasi lokasinya hanya dengan menuliskan alamat di form ataupun memberitahu secara lisan kepada pihak PDAM. Hal ini tentu akan mempersulit pihak PDAM dalam mencari alamat ketika lokasi kerusakan berada di pelosok dan tidak memiliki alamat yang jelas seperti nama jalan dan nomor rumah. Dikarenakan banyaknya pelanggan PDAM yang membutuhkan air bersih untuk kehidupan sehari-hari, sudah seharusnya PDAM memberikan pelayanan yang maksimal kepada pelanggannya.

Permasalahan mengenai pelaporan keluhan juga sudah dibahas dalam penelitian yang berjudul “Pengembangan Sistem Pengaduan Pelanggan Pada Perusahaan Daerah Air Minum Kota Salatiga” (Bintarawan, 2013). Pada penelitian ini dibangun sebuah sistem pengaduan pelanggan berbasis web. Pengguna dapat melihat informasi mengenai PDAM serta mengirimkan pengaduan dengan cara mengisi form pengaduan yang telah disediakan. Namun pada penelitian ini pengguna tidak dapat mengupload gambar terkait pengaduan yang dibuat. Selain itu juga tidak dapat memberikan lokasi dimana pelanggan tersebut berada sehingga ketika lokasi yang dituliskan pengguna pada form pengaduan kurang jelas maka akan mempersulit pihak PDAM untuk melakukan perbaikan.

Geotagging merupakan proses penambahan informasi yang berupa metadata geospasial pada berbagai macam media seperti gambar, video, website. Data yang diberikan yaitu berupa latitude dan longitude yang mencakup informasi lengkap suatu lokasi (Albharkah, 2016). Penerapan fitur *geotagging* juga sudah banyak digunakan dalam beberapa penelitian lainnya seperti pada penelitian dengan judul Rancang Bangun Aplikasi Mobile Social Geotagging Keragaman Budaya Indonesia Pada Platform Android (Ariyansyah, 2016). Pemanfaatan geotagging pada penelitian tersebut yaitu digunakan untuk menambahkan informasi lokasi pada foto keragaman budaya yang dilampirkan. Berdasarkan hasil pengujian didapatkan nilai SUS sebesar 82 yang berarti aplikasi dapat diterima dan mudah digunakan oleh pengguna.

Berdasarkan permasalahan yang telah diuraikan diatas, maka penulis melakukan penelitian dengan judul **“Pengembangan Aplikasi *Mobile* Pelaporan Keluhan Pelanggan PDAM menggunakan Fitur *Geotagging* Berbasis Android (Studi Kasus: PDAM Tirta Tuah Benua Kutai Timur)”**. Dimana pada aplikasi ini pelanggan dapat melaporkan keluhannya dengan mengisi form pengaduan serta menambahkan gambar yang terkait dengan keluhannya. Selain itu sistem ini juga memanfaatkan fitur geotagging yang memungkinkan pelanggan untuk memberikan informasi lokasinya secara tepat dan akurat sehingga pihak PDAM dapat dengan mudah mengetahui lokasi dari pelapor.

Dengan adanya aplikasi pelaporan keluhan pelanggan PDAM berbasis android ini diharapkan dapat membantu pelanggan untuk memberikan laporan keluhan mereka serta mempermudah PDAM untuk mengetahui lokasi akurat dari pelapor dengan menggunakan fitur geotagging sehingga PDAM dapat memberikan umpan balik terhadap laporan pelanggan dengan cepat dan tepat. Pelanggan juga dapat merasakan kemudahan ketika melaporkan keluhannya menggunakan sistem ini dibandingkan dengan cara sebelumnya sehingga dapat meningkatkan kepuasan pelanggan terhadap layanan PDAM.



## 1.2 Rumusan masalah

Berdasarkan penjabaran latar belakang diatas, maka didapatkan rumusan permasalahan sebagai berikut:

1. Bagaimana analisis dan perancangan aplikasi pelaporan keluhan pelanggan PDAM sehingga mampu membantu pelanggan PDAM dalam melaporkan keluhannya?
2. Bagaimana implementasi dari aplikasi pelaporan keluhan pelanggan PDAM sehingga sesuai dengan analisis serta perancangan yang telah dilakukan?
3. Bagaimana hasil pengujian aplikasi pelaporan keluhan pelanggan PDAM ditinjau dari pengujian perangkat lunak yang telah dilakukan?

## 1.3 Tujuan Penelitian

Penelitian ini memiliki tujuan untuk membangun suatu aplikasi pelaporan keluhan pelanggan PDAM guna menyelesaikan masalah pelaporan keluhan yang ada saat ini. Diharapkan dengan adanya penelitian ini dapat membantu masyarakat yang ingin melaporkan keluhannya. Selain itu penelitian ini juga memiliki tujuan khusus yaitu sebagai berikut:

1. Mengetahui analisis serta rancangan dari aplikasi pelaporan keluhan pelanggan PDAM yang dapat membantu pelanggan PDAM dalam melaporkan keluhannya
2. Mengetahui hasil implementasi dari aplikasi pelaporan keluhan pelanggan PDAM berdasarkan analisis dan perancangan yang telah dilakukan.
3. Mengetahui hasil pengujian aplikasi pelaporan keluhan pelanggan PDAM ditinjau dari pengujian *whitebox*, validasi, dan *compatibility* yang telah dilakukan.

## 1.4 Manfaat

Hasil dari penelitian ini nantinya diharapkan mampu memberikan beberapa manfaat sebagai berikut:

1. Manfaat bagi instansi, membantu meningkatkan pelayanan bagi pelanggan PDAM dengan memberikan tanggapan atas laporan yang telah dikirimkan.
2. Manfaat bagi peneliti, menambah wawasan mengenai perancangan serta implementasi dari sistem pelaporan keluhan pelanggan PDAM berbasis android
3. Manfaat bagi masyarakat, memberikan kemudahan serta kenyamanan bagi pelanggan PDAM untuk melaporkan keluhan atas masalah yang dialami kepada PDAM

## 1.5 Batasan masalah

Berdasarkan penjabaran rumusan masalah diatas, maka pada penelitian ini penulis membatasi pembahasan sebagai berikut:

1. Aplikasi yang akan dibangun berbasis android
2. Aplikasi yang akan dibangun membutuhkan perangkat dengan fitur GPS dan koneksi internet untuk dapat dijalankan
3. Bahasa pemrograman yang digunakan dalam penelitian ini yaitu Java, PHP, HTML, CSS
4. Informasi yang dikirimkan oleh pengguna yaitu laporan keluhan dalam bentuk *text*, gambar, serta lokasi dari pelanggan
5. Informasi lokasi yang ditambahkan berupa *latitude* dan *longitude*, bukan dalam bentuk *text*
6. Informasi yang diberikan kepada pengguna yaitu berupa *text* serta proses *event* dari kegiatan perbaikan yang sedang dikerjakan

## 1.6 Sistematika pembahasan

Bagian ini menjelaskan struktur dari skripsi ini mulai Bab 1 yaitu Pendahuluan sampai dengan Bab 7 yaitu Penutup yang dilengkapi dengan penjelasan singkat dari masing-masing bab. Dengan adanya bagian ini diharapkan sistematika pembahasan yang ada pada skripsi ini dapat dipahami oleh pembaca dengan baik.

Dalam penulisan penelitian ini, sistematika pembahasan yang digunakan yaitu

### **BAB 1 PENDAHULUAN**

Pada bab 1 ini menjelaskan mengenai Latar Belakang, Identifikasi Masalah, Rumusan Masalah, Tujuan Penelitian, Manfaat Penelitian, Batasan Penelitian, dan Sistematika Pembahasan.

### **BAB 2 LANDASAN KEPUSTAKAAN**

Pada bab 2 ini menjelaskan tentang teori-teori serta teknologi yang digunakan dan berhubungan dengan permasalahan yang diangkat dalam penelitian ini serta penelitian terdahulu yang dapat dijadikan sebagai pedoman pelaksanaan penelitian.

### **BAB 3 METODOLOGI PENELITIAN**

Pada bab 3 ini menjelaskan mengenai apa saja metode yang dilakukan dalam melaksanakan penelitian aplikasi pelaporan keluhan pelanggan PDAM berbasis android seperti studi literatur, analisis kebutuhan, metode pengambilan data, perancangan sistem, implementasi, pengujian dan pengambilan kesimpulan.

**BAB 4 ANALISIS KEBUTUHAN**

Pada bab 4 ini berisi analisis kebutuhan baik kebutuhan fungsional maupun non fungsional dari aplikasi pelaporan keluhan pelanggan PDAM berbasis android

**BAB 5 PERANCANGAN DAN IMPLEMENTASI**

Pada bab 5 ini menjelaskan perancangan serta bagaimana proses implementasi yang dilakukan pada aplikasi pelaporan keluhan pelanggan PDAM berbasis android yang dibuat. Implementasi ini meliputi Spesifikasi Sistem, serta Implementasi Antarmuka.

**BAB 6 PENGUJIAN DAN ANALISIS**

Pada bab 6 ini menjelaskan hasil dari beberapa pengujian yang telah dilakukan serta analisis pada sistem pelaporan keluhan pelanggan PDAM berbasis android.

**BAB 7 PENUTUP**

Pada bab 7 ini menjelaskan kesimpulan yang dapat diambil berdasarkan dari tahap analisis dan perancangan, tahap implementasi, serta tahap pengujian dan analisis yang dilakukan pada aplikasi pelaporan keluhan pelanggan PDAM berbasis android. Selain itu juga berisi saran yang dapat dilakukan pengembang apabila ingin melanjutkan penelitian ini lebih dalam.

## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Penelitian pertama yang menjadi acuan dalam penelitian ini yaitu penelitian yang berjudul “Pengembangan Sistem Pengaduan Pelanggan pada Perusahaan Daerah Air Minum Kota Salatiga” yang dilakukan oleh Widya Ari Bintarawan. Pada penelitian ini dibuat sebuah sistem berbasis web yang dapat membantu pelanggan PDAM dalam mengirimkan laporan pengaduan secara online melalui web. Kemudian laporan yang dikirimkan oleh pelanggan akan diolah sehingga menghasilkan *report* laporan pengaduan dalam bentuk grafik yang dapat dicetak. *Report* ini akan membantu perusahaan dalam mengambil kebijakan. Berdasarkan pengujian yang dilakukan, sistem yang dibangun sangat membantu pelanggan PDAM dalam melakukan pengaduan kepada perusahaan (Bintarawan, 2013). Perbedaan penelitian tersebut dengan penelitian yang akan penulis buat yaitu sistem yang dibuat berbasis web dan tidak menggunakan fitur *geotagging*. Sedangkan aplikasi yang akan penulis buat berbasis Android dan menggunakan fitur *geotagging*.

Penelitian selanjutnya yaitu penelitian dengan judul “Rancang Bangun Aplikasi *Geotagging Social Report* Bencana Banjir”. Pada penelitian ini teknologi *geotagging* untuk melaporkan adanya banjir yang akan disimpan dalam bentuk titik-titik lokasi yang nantinya ditampilkan dalam peta kejadian bencana banjir. Berdasarkan pengujian yang dilakukan, aplikasi yang telah dibangun dapat dengan mudah digunakan oleh masyarakat (Wijaya, et al., 2018). Perbedaan dengan penelitian yang penulis buat yaitu penulis membuat aplikasi yang ditujukan untuk melaporkan keluhan pelanggan PDAM.

Penelitian selanjutnya yaitu penelitian yang dilakukan oleh Mochamad Halili MZ yang berjudul “Aplikasi Rekomendasi Spot Area Wisata Berbasis Android dengan Teknik Geotag”. Pada penelitian ini menggunakan metode *Location Based Service* untuk mengidentifikasi lokasi *user* serta menggunakan fitur *geotagging* untuk menyajikan informasi tempat wisata kepada *user* (MZ, 2017). Perbedaan penelitian tersebut dengan penelitian yang akan penulis buat yaitu, penulis menggunakan fitur *geotagging* untuk mengirimkan gambar lokasi dari laporan yang dibuat.

Penelitian keempat yaitu penelitian yang dilakukan oleh Eka Ariyansyah dengan judul “Rancang Bangun Aplikasi Mobile Social Geotagging Keragaman Budaya Indonesia Pada *Platform* Android”. Pada penelitian ini dibuat sebuah sistem untuk mempromosikan keragaman budaya secara *online* menggunakan metode *crowdsourcing* dan *geotagging*. Metode *geotagging* pada penelitian ini digunakan untuk memberikan informasi lokasi mengenai budaya yang di berikan oleh masyarakat dengan menggunakan konsep *crowdsourcing*. Berdasarkan pengujian yang dilakukan, didapatkan hasil bahwa sistem yang dibangun dapat diterima dengan baik oleh masyarakat (Ariyansyah, 2016). Perbedaan penelitian ini dengan penelitian yang akan penulis buat yaitu tidak digunakannya konsep



*crowdsourcing* dan penerapan geotagging diimplementasikan pada aplikasi pelaporan keluhan.

Penelitian terakhir yang menjadi acuan dalam penelitian ini yaitu penelitian yang dilakukan oleh Fahmi Hammadi Wiriyo dengan judul “Pengembangan Aplikasi Mobile Pelaporan Keluhan Pelanggan Listrik Menggunakan Fitur *Location Based Services* Berbasis Android”. Pada penelitian ini, pencarian lokasi keluhan pelanggan menggunakan fitur Location Based Service agar lebih akurat dengan memanfaatkan fitur GPS pada *smartphone*. Kemudian aplikasi dapat memberikan umpan balik terhadap laporan keluhan oleh pelanggan (Wiriyo, 2016). Perbedaan penelitian yang akan penulis buat dengan penelitian tersebut yaitu aplikasi yang dibuat akan diimplementasikan pada Perusahaan Daerah Air Minum. Serta digunakannya fitur geotagging untuk mengidentifikasi lokasi pelanggan.

**Tabel 2.1 Kajian Pustaka**

No	Judul	Penulis	Perbandingan	
			Kajian Pustaka	Penelitian Penulis
1	Pengembangan Sistem Pengaduan Pelanggan pada Perusahaan Daerah Air Minum Kota Salatiga	(Bintarawan, 2013)	Sistem yang dibuat berbasis web dan tidak menggunakan fitur <i>geotagging</i>	Sistem yang dibuat berbasis android dan menggunakan fitur <i>geotagging</i>
2	Rancang Bangun Aplikasi <i>Geotagging Social Report</i> Bencana Banjir	(Wijaya, et al., 2018)	Implementasi fitur <i>geotagging</i> digunakan untuk melaporkan bencana banjir yang dapat dilihat pengguna lain	Implementasi fitur <i>geotagging</i> digunakan untuk memberikan informasi lokasi pada keluhan pelanggan PDAM
3	Aplikasi Rekomendasi Spot Area Wisata Berbasis Android dengan Teknik <i>Geotag</i>	(MZ, 2017)	Implementasi fitur <i>geotagging</i> digunakan untuk menyajikan informasi wisata kepada pengguna dan <i>Location Based Service</i> untuk mengidentifikasi lokasi pengguna	Implementasi fitur <i>geotagging</i> digunakan untuk memberikan informasi lokasi pada keluhan pelanggan PDAM



**Tabel 2.1 Kajian Pustaka (Lanjutan)**

No	Judul	Penulis	Perbandingan	
			Kajian Pustaka	Penelitian Penulis
4	Rancang Bangun Aplikasi <i>Mobile Social Geotagging</i> Keragaman Budaya Indonesia Pada <i>Platform</i> Android	(Ariyansyah, 2016)	Implementasi fitur <i>geotagging</i> digunakan untuk memberikan informasi budaya dengan menggunakan konsep <i>crowdsourcing</i>	Implementasi fitur <i>geotagging</i> digunakan untuk memberikan informasi lokasi pada keluhan pelanggan PDAM
5	Pengembangan Aplikasi Mobile Pelaporan Keluhan Pelanggan Listrik Menggunakan Fitur <i>Location Based Services</i> Berbasis Android	(Wiriyo, 2016)	Implementasi <i>Location Based Service</i> untuk mengidentifikasi lokasi pelanggan PLN	Implementasi fitur geotagging digunakan untuk memberikan informasi lokasi pada keluhan pelanggan PDAM

## 2.2 Pelaporan Keluhan Pelanggan

Pelaporan keluhan pelanggan pada PDAM Tirta Tuah Benua Kutai Timur secara umum ditangani oleh Bagian Hubungan Langsung (Hublang). Pada proses penanganannya laporan pelanggan dibagi menjadi 4 jenis laporan berbeda dimana masing-masing laporan akan ditangani oleh bagian yang berbeda. Empat bagian tersebut yaitu:

1. **Distribusi**, pada bagian ini akan melayani laporan keluhan pelanggan yang berupa masalah teknis seperti adanya kebocoran pipa atau pipa rusak yang diakibatkan oleh berbagai faktor seperti bencana alam ataupun kecelakaan.
2. **Produksi**, pada bagian ini akan melayani laporan keluhan pelanggan yang berupa masalah kualitas, kontinuitas, dan kuantitas air yang diterima oleh pelanggan. Masalah kualitas yaitu apabila air dalam kondisi yang tidak baik seperti berbau tidak sedap, berbuih atau keruh. Masalah kontinuitas yaitu apabila air tidak mengalir setiap hari. Sedangkan masalah kuantitas yaitu apabila air mengalir namun dengan kuantitas yang kecil.
3. **Perencanaan**, pada bagian ini akan melayani laporan keluhan pelanggan yang ingin membuat sambungan baru atau memindahkan lokasi meteran air.

4. **Kas dan Penagihan**, pada bagian ini akan melayani laporan keluhan pelanggan yang berkaitan dengan tagihan air, seperti tagihan yang melebihi pemakaian.

Pada proses penanganan laporan pelanggan terdapat 3 status penanganan yang dilalui pelanggan ketika melaporkan keluhan, antara lain:

1. Belum Ditangani

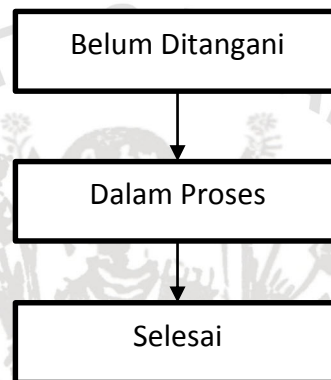
Pada tahap ini laporan pelanggan dalam proses pendataan dan belum ditangani lebih lanjut.

2. Dalam proses

Pada tahap ini laporan pelanggan dalam proses pengerjaan atau perbaikan.

3. Selesai

Pada tahap ini laporan pelanggan telah selesai ditangani.



**Gambar 2.1 Diagram Alur Status Penanganan Keluhan Pelanggan**

### 2.3 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah proses membangun sebuah perangkat lunak yang menerapkan rekayasa sebagai konsep sehingga perangkat lunak yang dihasilkan dapat memiliki nilai ekonomi dan secara efisien bekerja dengan menggunakan mesin (S & Shalahuddin, 2016). Rekayasa perangkat lunak dibutuhkan agar perangkat lunak yang dikembangkan ataupun dibangun telah sesuai dengan apa yang pengguna butuhkan sehingga dapat terpakai dengan maksimal. Rekayasa perangkat lunak akan berfokus pada pembuatan perangkat lunak dengan kriteria yang dapat terus dipelihara seiring dengan berkembangnya teknologi dan zaman, dapat diandalkan terhadap proses bisnis yang berjalan, efisien pada segi penggunaannya serta dapat dipakai sesuai kebutuhannya.

Secara umum, ada beberapa tahap yang harus dilakukan pengembang pada saat proses rekayasa perangkat lunak, antara lain (S & Shalahuddin, 2016):

1. Analisis

Tahap analisis merupakan tahap untuk mengidentifikasi kebutuhan-kebutuhan apa saja yang dibutuhkan sistem untuk dapat menyelesaikan

masalah dari pengguna. Selain itu juga untuk mengidentifikasi siapa saja aktor yang terlibat dengan sistem.

## 2. Perancangan

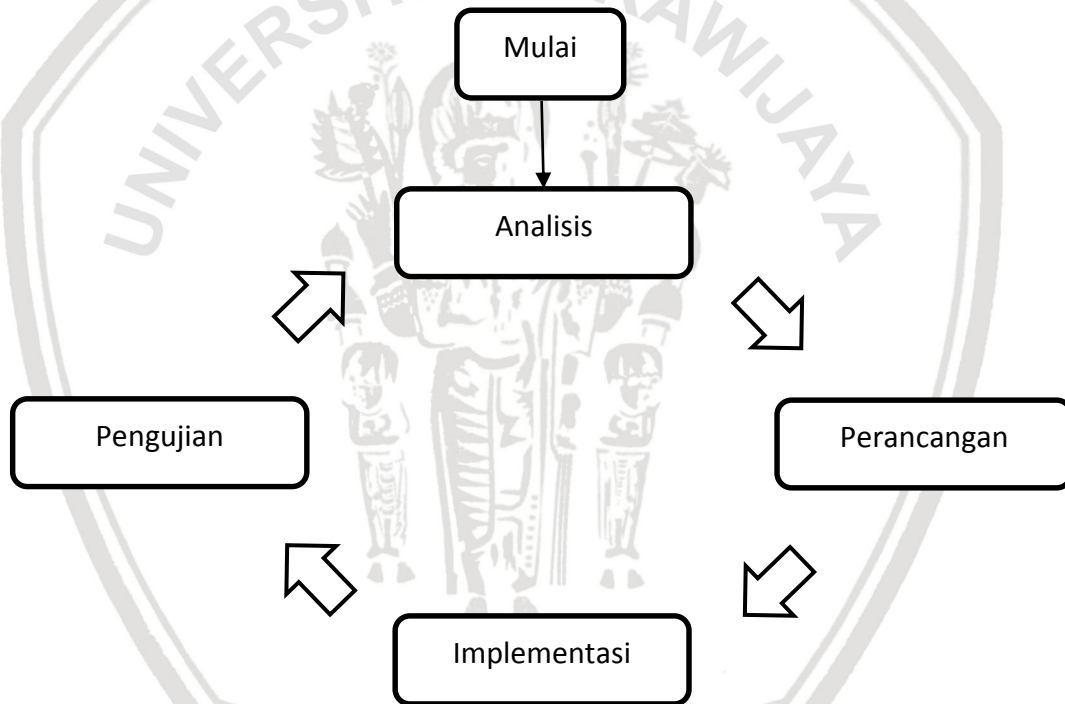
Pada tahap ini, akan dibuat perancangan dari sebuah sistem berdasarkan analisis yang telah dilakukan sebelumnya. Perancangan akan dibuat acuan untuk tahapan implementasi sistem.

## 3. Implementasi

Tahap implementasi adalah tahap untuk merealisasikan rancangan sistem yang sebelumnya telah dibuat. Pada tahap ini akan dibangun sistem sesuai dengan rancangan yang telah ada.

## 4. Pengujian

Setelah implementasi sistem telah selesai, akan dilakukan pengujian apakah sistem yang dibangun telah sesuai dengan hasil analisis serta perancangan yang telah dibuat.



**Gambar 2.2 Tahapan Umum Rekayasa Perangkat Lunak**

Sumber: (S & Shalahuddin, 2016)

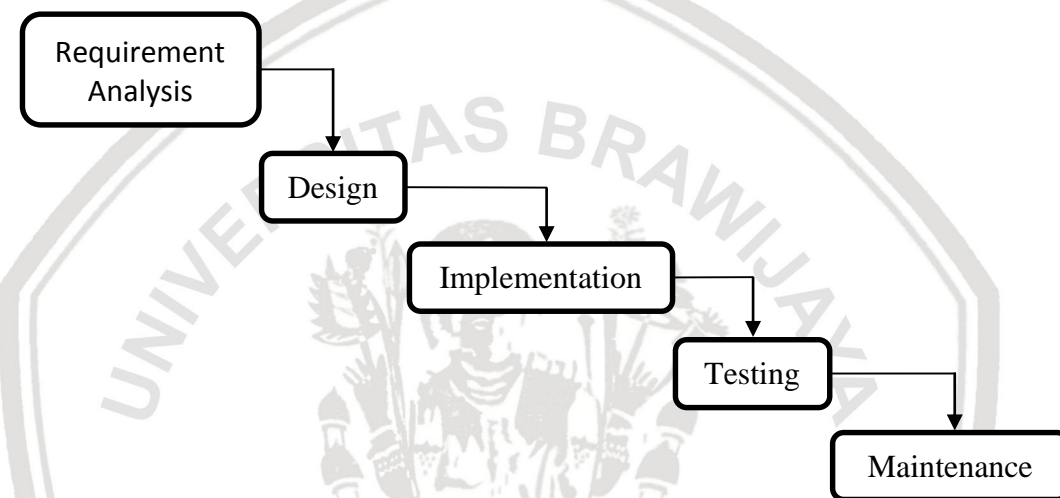
## 2.4 Software Development Life Cycle

*Software Development Life Cycle* (SDLC) merupakan proses dalam pengembangan perangkat lunak yang dilalui pengembang dengan menggunakan metodologi dan model-model yang telah digunakan dalam pengembangan perangkat lunak sebelumnya (S & Shalahuddin, 2016). Terdapat beberapa model

SDLC antara lain *waterfall*, *prototype*, *Rapid Application Development (RAD)*, *iterative*, dan *spiral*.

### 2.4.1 Waterfall Model

Model SDLC *waterfall* (air terjun) atau sering disebut dengan *sequential linear* adalah salah satu dari beberapa model SDLC klasik dimana alur hidup dari pengembangan perangkat lunak digambarkan menggunakan pendekatan sekuensial atau terurut dimulai dari analisis, desain, membuat kode, pengujian serta tahap pendukung (*support*) (S & Shalahuddin, 2016). Pada model *waterfall* proses pengembangan perangkat lunak dapat berjalan dengan baik apabila dokumentasi serta perencanaan dilakukan dengan baik. Pada Gambar 2.3 akan digambarkan diagram alir *waterfall* model



**Gambar 2.3 Diagram Alir Waterfall Model**

Sumber: (S & Shalahuddin, 2016)

Salah satu alasan model *waterfall* sering digunakan oleh pengembang perangkat lunak yaitu karena cocok dan mudah digunakan untuk mengembangkan perangkat lunak dengan skala kecil dengan kebutuhan pelanggan yang sudah dipahami dengan baik. Selain itu tahapan pengembangan dilakukan dengan jelas serta di dokumentasikan dengan jelas (S & Shalahuddin, 2016).

### 2.5 Unified Modeling Language

*Unified Modeling Language (UML)* yaitu sebuah standarisasi Bahasa pemodelan yang digunakan untuk mengembangkan perangkat lunak dengan menggunakan pemrograman yang berorientasi objek (S & Shalahuddin, 2016). UML digunakan untuk merepresentasikan sebuah perangkat lunak ke dalam bentuk pemodelan visual. Terdapat beberapa diagram yang termasuk dalam UML yang digunakan dalam pembangunan aplikasi pelaporan keluhan, antara lain *use case diagram* dan *use case skenario*.


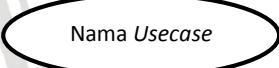

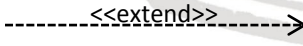
### 2.5.1 Use Case Diagram

*Use case* diagram adalah sebuah diagram yang merepresentasikan hubungan dan interaksi antara sebuah sistem yang dibangun dengan satu atau lebih aktor yang terlibat. Dengan kata lain sebuah *use case* diagram akan menggambarkan apa saja fungsi yang dapat dilakukan oleh sistem untuk dapat memenuhi kebutuhan pengguna dalam menyelesaikan masalah serta siapa saja yang dapat melakukan fungsi-fungsi tersebut. Terdapat dua komponen utama di dalam *use case* diagram yaitu aktor dan *use case*.

1. Aktor, merupakan suatu orang atau sistem lain yang melakukan interaksi dengan perangkat lunak yang akan dibangun. Aktor digambarkan dengan simbol orang
2. *Use case*, merupakan fungsionalitas yang dapat dilakukan oleh aktor. *Use case* digambarkan dengan simbol elips

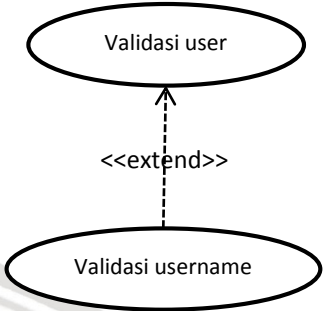

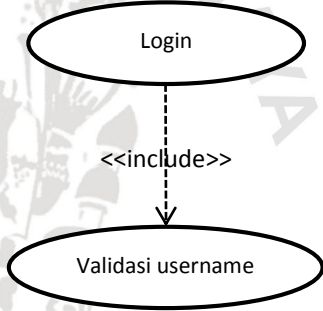

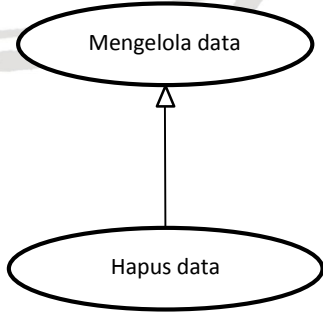
Pada Tabel 2.2 akan dijelaskan simbol-simbol yang terdapat pada *use case* diagram.

**Tabel 2.2 Simbol pada Use Case Diagram**

Simbol	Deskripsi
Aktor 	Orang atau sistem lain yang memiliki interaksi dengan perangkat lunak yang akan dibangun
<i>Use Case</i> 	Fungsionalitas yang dimiliki oleh sistem yang akan dibangun dan saling bertukar pesan antar unit ataupun aktor
Asosiasi 	Hubungan antara aktor dan <i>use case</i> yang berinteraksi dengan suatu <i>use case</i> atau sebuah <i>use case</i> yang memiliki interaksi dengan aktor
<i>Extend</i> 	Hubungan antar dua <i>use case</i> , dimana <i>use case</i> yang ditambahkan tetap bisa berdiri sendiri walaupun tanpa adanya <i>use case</i> yang ditambahkan tersebut. Arah panah menuju pada <i>use case</i> yang ditambahkan.



Tabel 2.2 Simbol pada *Use Case Diagram* (Lanjutan)

Simbol	Deskripsi
	<p>Contoh:</p> 
<p><i>Include</i></p> 	<p>Hubungan antar dua <i>use case</i>, dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> tambahan tersebut untuk dapat menjalankan fungsinya. Dengan kata lain <i>use case</i> yang ditambahkan akan selalu dipanggil ketika <i>usecase</i> tambahan akan dijalankan</p> <p>Contoh:</p> 
<p>Generalisasi</p> 	<p>Hubungan antar dua buah <i>use case</i> dimana suatu fungsi merupakan fungsi yang lebih umum dari fungsi yang lainnya. Arah panah menu pada <i>use case</i> yang menjadi generalisasinya atau yang lebih umum.</p> <p>Contoh:</p> 


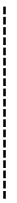

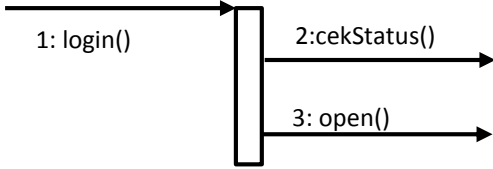
Sumber: (S & Shalahuddin, 2016)

### 2.5.2 Sequence Diagram

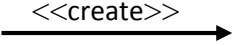
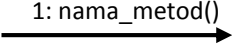
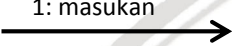
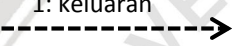
*Sequence* Diagram adalah sebuah diagram yang merepresentasikan aktivitas suatu objek terhadap sebuah *use case* dengan mendeskripsikan pertukaran *message* antar objek tersebut yang di tampilkan dalam urutan waktu (S & Shalahuddin, 2016). Pembuatan *sequence* diagram mengacu pada *use case* skenario yang telah dibuat sebelumnya. Sebelum membuat sebuah *sequence* diagram, objek-objek yang terlibat dalam sebuah *use case* harus diidentifikasi terlebih dahulu beserta metod-metod yang dimiliki dari sebuah kelas yang diinstansiasi menjadi objek tersebut.

Pada Tabel 2.3 akan dijelaskan simbol yang terdapat pada *Sequence* Diagram beserta deskripsinya.

**Tabel 2.3 Simbol pada *Sequence* Diagram**

Simbol	Deskripsi
<p>Aktor</p>  <p>Nama Aktor</p>	<p>Orang atau sistem lain yang melakukan interaksi dengan perangkat lunak yang akan dibangun</p>
<p>Garis Hidup/<i>LifeLine</i></p> 	<p>Menggambarkan garis hidup/kehidupan dari suatu objek</p>
<p>Waktu Aktif</p> 	<p>Menggambarkan waktu aktif dari sebuah objek dimana objek tersebut dapat berinteraksi dengan objek lain. Semua yang berhubungan dengan waktu aktif merupakan tahapan yang sedang dilakukan didalamnya</p> <p>Contoh:</p>  <p>Maka cekStatus() dan open() dilakukan didalam metod login()</p>

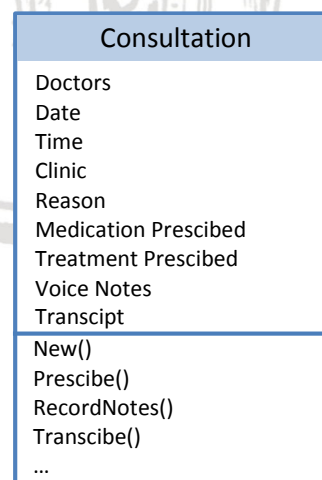
**Tabel 2.3 Simbol pada *Sequence Diagram* (Lanjutan)**

Simbol	Deskripsi
Pesan tipe <i>create</i> 	Menggambarkan bahwa suatu objek membuat objek yang lain dengan arah panah menuju ke objek yang akan dibuat
Pesan tipe <i>call</i> 	Menggambarkan bahwa suatu objek sedang memanggil metod yang terdapat di objek lainnya atau diri sendiri dengan arah panah menuju ke objek yang memiliki metod tersebut.
Pesan tipe <i>send</i> 	Menggambarkan bahwa suatu objek mengirimkan data ataupun informasi ke objek lainnya dengan arah panah mengarah ke objek yang dikirim
Pesan tipe <i>return</i> 	Menggambarkan bahwa suatu objek yang telah menjalankan sebuah metod telah menghasilkan kembalian menuju objek tertentu dengan arah panah mengarah pada objek yang diberi kembalian

Sumber: (S & Shalahuddin, 2016)

### 2.5.3 *Class Diagram*

*Class Diagram* adalah suatu diagram yang digunakan untuk menggambarkan kelas-kelas yang ada dalam pembuatan sebuah sistem. Di dalam *class diagram* terdapat atribut, operasi/metod, serta relasi antar kelas tersebut (Sommerville, 2011). Pada Gambar 2.4 akan ditunjukkan contoh dari *class diagram* beserta atribut dan methodnya.



**Gambar 2.4 Contoh *Class Diagram***

Sumber: (Sommerville, 2011)

## 2.6 Android

Android merupakan platform pengembangan perangkat lunak untuk perangkat bergerak (*mobile devices*) yang dikembangkan oleh Google. Sistem operasi yang digunakan pada Android yaitu Sistem Operasi berbasis Linux yang telah dimodifikasi dan terdiri dari sistem operasi, *middleware*, serta beberapa aplikasi utama. Android merupakan OS pertama yang bersifat *open source* dimana dalam pengembangannya tidak hanya dalam ruang lingkup sistem operasi tetapi juga mengembangkan aplikasi menggunakan (*Software Development Kit*) SDK Android serta menggunakan Bahasa pemrograman Java (Istiyanto, 2013).

Terdapat empat komponen yang ada pada Aplikasi Android, antara lain (Huda, 2012):

1. *Activity*, merupakan bagian dari aplikasi yang digunakan untuk berinteraksi antara pengguna dan aplikasi. *Activity* memiliki *user interface* untuk dapat digunakan oleh pengguna
2. *Service*, merupakan komponen yang berjalan pada background sehingga tidak memiliki *user interface*.
3. *Broadcast Receiver*, merupakan komponen yang digunakan untuk menerima kode dari sistem Android.
4. *Content Provider*, merupakan komponen yang mengizinkan data dari sebuah aplikasi untuk dapat diakses oleh aplikasi lainnya.

### 2.6.1 Versi Android

Semakin majunya perkembangan teknologi saat ini tentu diikuti pula dengan berkembangnya versi Sistem Operasi Android yang ada. Berbagai macam versi baru OS Android diluncurkan guna memuaskan kebutuhan pasar global yang semakin hari semakin meningkat (Firly, 2018). Pada Tabel 2.4 digambarkan berbagai versi Android yang telah dirilis oleh Android.

**Tabel 2.4 Versi Android**

Versi	Nama	Tanggal Rilis
1.0 (API level 1)	-	23 September 2008
1.1 (API level 2)	-	9 Februari 2009
1.5 (API level 3)	Cupcake	27 April 2009
1.6 (API level 4)	Donut	15 September 2009
2.0 (API level 5)	Éclair	26 Oktober 2009
2.0.1 (API level 6)	Éclair	3 Desember 2009
2.1 (API level 7)	Éclair	12 Januari 2010
2.2 – 2.2.3 (API level 8)	Froyo	20 Mei 2010

Tabel 2.4 Versi Android (Lanjutan)

Versi	Nama	Tanggal Rilis
2.3 – 2.3.2 (API level 9)	Gingerbread	6 Desember 2010
2.3.3 – 2.3.7 (API level 10)	Gingerbread	9 Februari 2011
3.0 (API level 11)	Honeycomb	22 Februari 2011
3.1 (API level 12)	Honeycomb	10 Mei 2011
3.2 (API level 13)	Honeycomb	15 Juli 2011
4.0 – 4.0.2 (API level 14)	Ice Cream Sandwich	19 Oktober 2011
4.0.3 – 4.0.4 (API level 15)	Ice Cream Sandwich	16 Desember 2011
4.1 (API level 16)	Jelly Bean	27 Juni 2012
4.2 (API level 17)	Jelly Bean	29 Oktober 2012
4.3 (API level 18)	Jelly Bean	24 Juli 2013
4.4 (API level 19)	KitKat	31 Oktober 2013
5.0 (API level 21)	Lollipop	12 November 2014
6.0 (API level 23)	Marshmallow	5 Oktober 2015
7.0 (API level 24)	Nougat	9 Maret 2016
7.1 (API level 25)	Nougat	19 Oktober 2016
8.0 (API level 26)	Oreo	21 Maret 2017

Sumber: (Firly, 2018)

## 2.7 Website

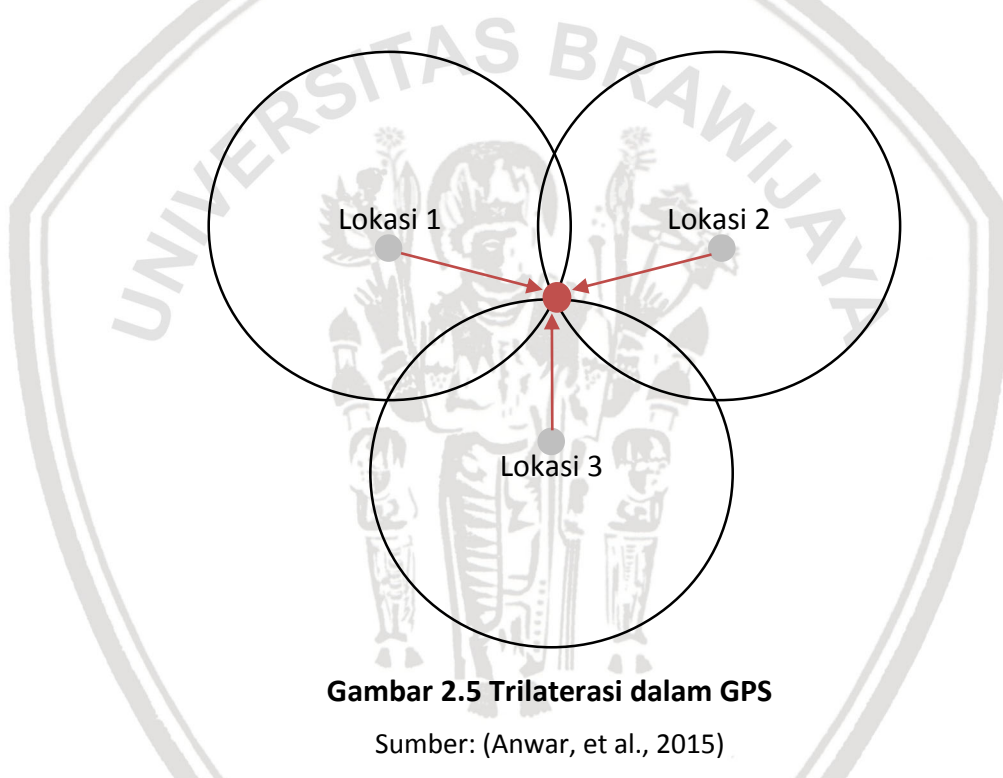
*Website* merupakan sekumpulan halaman yang terdapat pada sebuah domain dan memiliki hubungan dengan berbagai *file* lain yang saling terkait. Hubungan antar halaman web tersebut biasa disebut dengan *Hyperlink* sedangkan media penghubung yang berupa teks disebut dengan *Hypertext*. Domain merupakan nama unik yang dimiliki sebuah institusi untuk dapat diakses melalui sebuah internet seperti yahoo.com, turnitin.com dan lain-lain (Yuhefizar, 2013).

Dalam sebuah *website* terdapat satu halaman yang akan dilihat pertama ketika seseorang mengakses sebuah *website*, halaman tersebut sering disebut dengan *home page*. Ketika pengguna akan mengakses sebuah *website*, pengguna harus memasukkan alamat URL (*Unified Resource Locator*) pada kolom URL yang telah ada pada sebuah web *browser*. Ketika seorang pengguna memasukkan alamat URL maka web *browser* akan mengirimkan *request* pada TCP/IP kemudian akan dilakukan pemeriksaan apakah permintaan pengguna telah sesuai dengan alamat yang terdapat pada *server* atau tidak lalu *server* akan mengirimkan hasil kepada pengguna berdasarkan pemeriksaan yang telah dilakukan melalui alamat IP-nya (Fauzia, 2017).



## 2.8 GPS

*Global Positioning Sistem* (GPS) merupakan sebuah sistem navigasi yang berbasis satelit dan saling berhubungan yang berada pada orbitnya (Istiyanto, 2013). Sistem GPS merupakan konvergensi dari beberapa perangkat komunikasi nirkabel yang menggunakan jalur frekuensi tinggi sehingga satelit GPS dapat berkomunikasi dengan *receiver* GPS secara langsung tanpa harus dipengaruhi oleh faktor geografis serta cuaca (Istiyanto, 2013). Untuk dapat berkomunikasi dengan *receiver*, GPS akan mengirimkan sinyal berupa gelombang mikro ke bumi yang nantinya akan diterima oleh *receiver* yang berada di permukaan bumi. *Receiver* ini yang akan mengumpulkan informasi dari GPS berupa waktu, lokasi, kecepatan, arah perjalanan, simpan lokasi, serta komulasi data (Anwar, et al., 2015). Untuk dapat menjaga akurasi data yang dikirimkan pada *receiver* maka satelit harus berada pada orbit yang tepat.



**Gambar 2.5 Trilaterasi dalam GPS**

Sumber: (Anwar, et al., 2015)

Berdasarkan Gambar 2.5, sebuah GPS *receiver* dapat menerima informasi dari tiga ataupun lebih satelit. GPS *receiver* harus berada pada jangkauan dari semua satelit yang berhubungan dengan GPS *receiver* tersebut. Serta satelit-satelit tersebut harus selalu berada pada posisi yang dapat menjangkau semua area di permukaan bumi sehingga mengurangi adanya *blank spot*. GPS *receiver* berisi beberapa *integerated circuit* (IC) sehingga dapat dengan mudah digunakan serta memiliki harga yang murah. GPS dapat dimanfaatkan pada berbagai bidang seperti pada pertanian, mobil, pesawat terbang serta dapat diintegrasikan dengan tablet, computer maupun *smartphone*.

## 2.9 Google Map API

Google Maps adalah sebuah layanan yang menyajikan peta dunia secara virtual yang telah disediakan oleh Google Inc (Minarni & Yusdi, 2015). Google Maps dapat kita akses melalui alamat <http://maps.google.com>. Dengan menggunakan Google Maps kita dapat melihat suatu daerah di dunia dengan hanya mengakses sebuah browser. Informasi yang disediakan oleh Google Maps berupa informasi geografis seperti nama jalan, nama gedung/fasilitas umum seperti hotel, ATM dan Pom bensin.

Untuk dapat menghubungkan suatu perangkat lunak seperti android dengan Google Maps diperlukan sebuah kunci yang disebut dengan API Key, selain itu juga diperlukan pustaka eksternal yang telah disediakan google yaitu [com.google.android.maps](http://com.google.android.maps). Untuk memanfaatkan pustaka tersebut diperlukan *API's add-on* sehingga aplikasi android yang dibuat dapat mengakses google maps.

## 2.10 Geotagging

*Geotagging* adalah proses penambahan data dalam bentuk metadata berupa informasi posisi *longitude*, *latitude* serta *altitude* yang didapatkan dari GPS kedalam sebuah media baik foto maupun video. Untuk melakukan *geotagging* pada suatu media, terdapat tiga cara yang berbeda, antara lain (Nandipati, 2011):

1. *Geocoding* manual, yaitu memasukkan koordinat suatu lokasi secara manual atau memilih suatu lokasi dengan manual ketika mengunggah suatu media ke dalam internet. Tingkat ketepatan akurasi lokasi dari *geotagging* manual ini tergantung pada GPS *receiver* atau *tools* yang digunakan untuk memperoleh koordinat.
2. Kamera digital yang memiliki fitur GPS didalamnya, pada beberapa kamera digital telah dilengkapi oleh fitur GPS secara langsung sehingga pada EXIF *header* hasil foto atau video yang diambil akan otomatis menyimpan informasi lokasi.
3. Kamera digital dengan fitur GPS yang terpisah, untuk menambahkan informasi lokasi pada EXIF *header* hasil foto atau video dapat menggunakan sebuah perangkat lunak untuk menyinkronkan kamera dengan GPS sehingga koordinat dapat direkam GPS pada waktu pengambilan foto dengan kamera.

Pada penelitian yang penulis buat, akan digunakan cara kedua yaitu gambar yang telah diambil melalui kamera smartphone yang dilengkapi dengan GPS akan otomatis memiliki informasi lokasi pada EXIF *header*-nya.

## 2.11 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan suatu tahap untuk memastikan perangkat lunak yang telah dibangun telah sesuai dengan fungsionalitas yang dirancang sebelumnya. Pengujian dilakukan dalam rangka untuk mengurangi

tingkat kesalahan ataupun kekurangan pada sebuah perangkat lunak (Simarmata, 2010).

### 2.11.1 Pengujian *Whitebox*

Pengujian *whitebox* adalah suatu metode pengujian yang berfokus pada detail perancangan program dari sistem yang dibangun (Mustaqbal, et al., 2015). Pada pengujian *whitebox* akan dibuat beberapa kasus uji dari struktur kontrol perancangan program. Pengujian ini bertujuan menghasilkan program yang benar 100% dengan cara memeriksa setiap fungsi kode program dengan beberapa kasus uji. Beberapa kasus uji yang digunakan memiliki beberapa tahapan, antara lain (Mustaqbal, et al., 2015):

1. Pengujian pada semua keputusan yang menggunakan logical
2. Pengujian pada struktur data yang terjamin validitasnya
3. Pengujian pada seluruh perulangan dengan sesuai pada batasan-batasan yang ada

### 2.11.2 Pengujian *Blackbox*

Pengujian *Blackbox* merupakan salah satu teknik pengujian yang berfokus pada kebutuhan fungsionalitas sistem, dan tidak berfokus pada alur kode program. Pada pengujian ini, penguji mendefinisikan beberapa *testcase* dan ditentukan kondisi inputnya untuk kemudian dilakukan pengujian pada *testcase* tersebut. Pengujian *blackbox* dapat melengkapi pengujian *whitebox* dengan menguji hal-hal yang tidak tercakup pada pengujian *whitebox*. Pengujian *blackbox* dilakukan untuk menemukan hal sebagai berikut (Mustaqbal, et al., 2015):

1. Fungsi yang tidak ada atau yang salah
2. Kesalahan yang ada pada antarmuka
3. Kesalahan pada akses *database* dan struktur data
4. Kesalahan yang ada pada performa

### 2.11.3 Pengujian Validasi

Pengujian validasi merupakan suatu strategi pengujian yang termasuk kedalam pengujian *blackbox*. Pada pengujian validasi ini akan berfokus untuk menguji apakah kebutuhan yang didefinisikan sebelumnya telah diimplementasikan dengan baik pada perangkat lunak yang dibangun. Untuk mengetahui apakah ada kesalahan, akan dibuat sebuah kasus uji, kemudian dari kasus uji tersebut dilakukan pengujian dengan menjalankan perangkat lunak dan membandingkan antara harapan yang diinginkan dengan *output* yang dihasilkan ketika perangkat lunak dijalankan. Apabila semua fungsi pada perangkat lunak yang dilakukan pengujian validasi dapat berjalan sesuai dengan yang diharapkan pada saat proses analisis dan perancangan yang telah dilakukan maka pengujian tersebut dapat dikatakan berhasil (Pressman & Maxim, 2015).

#### 2.11.4 Pengujian *Compatibility*

Pengujian *compatibility* yaitu suatu pengujian yang dilakukan untuk menguji kebutuhan non fungsional dari sebuah perangkat lunak. Pengujian ini dilakukan untuk memastikan apakah perangkat lunak yang dibangun dapat berjalan dengan baik pada beberapa sistem operasi, perangkat keras atau lingkungan yang berbeda. Pengujian *compatibility* pada aplikasi berbasis *mobile* dapat dilakukan dengan menguji aplikasi pada device yang berbeda dengan versi OS yang berbeda, ukuran layar serta resolusi yang berbeda (Kumar, et al., 2015).



### BAB 3 METODOLOGI

Metode penelitian yang digunakan dalam penelitian ini dibagi menjadi beberapa tahap, antara lain Studi Literatur, Analisis Kebutuhan, Pengumpulan Data, Perancangan Sistem, Implementasi, Pengujian dan Analisis, serta Kesimpulan. Diagram Alir Metodologi Penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian



### 3.1 Studi Literatur

Studi Literatur yang dilakukan pada penelitian ini yaitu dengan cara mengumpulkan data pendukung yang terkait dengan permasalahan yang diangkat serta dapat dijadikan referensi. Data dapat diambil dari beberapa sumber seperti jurnal yang telah diterbitkan maupun website resmi. Data yang dimaksud antara lain adalah dasar-dasar teori yang sesuai serta penelitian terdahulu.

Penelitian sebelumnya yang dapat dilakukan sebagai acuan yaitu berbagai penelitian yang memiliki kesamaan dengan penelitian yang akan penulis lakukan baik itu dari segi teknologi, topik, ataupun metode yang akan digunakan.. Lalu pada penelitian tersebut akan dicari keterikatan dengan penelitian yang sedang dilakukan serta bagaimana kelebihan dan kekurangan yang dapat diperbaiki. Adapun berbagai hal yang dijadikan Studi Literatur dari penelitian yang akan penulis lakukan yaitu:

1. Rekayasa Perangkat Lunak
2. *Software Development Life Cycle*
3. *Waterfall Model*
4. *Unified Modeling Language*
5. *Use Case Diagram*
6. *Sequence Diagram*
7. *Class Diagram*
8. Android
9. *Website*
10. GPS
11. Google Maps API
12. *Geotagging*
13. Pengujian Perangkat Lunak
14. Pengujian *Whitebox*
15. Pengujian *Blackbox*
16. Pengujian Validasi
17. Pengujian *Compatibility*
18. Jurnal Penelitian dengan judul “Pengembangan Sistem Pengaduan Pelanggan pada Perusahaan Daerah Air Minum Kota Salatiga”
19. Jurnal Penelitian dengan judul “Rancang Bangun Aplikasi *Geotagging Social Report* Bencana Banjir”
20. Jurnal Penelitian dengan judul “Aplikasi Rekomendasi Spot Area Wisata Berbasis Android dengan Teknik *Geotag*”

21. Jurnal Penelitian dengan judul “Rancang Bangun Aplikasi Mobile Social *Geotagging* Keragaman Budaya Indonesia Pada Platform Android”
22. Jurnal Penelitian dengan judul “Pengembangan Aplikasi Mobile Pelaporan Keluhan Pelanggan Listrik Menggunakan Fitur *Location Based Services* Berbasis Android”

### 3.2 Pengumpulan Data

Pada tahap ini akan dilakukan pengumpulan data sampel di PDAM Tirta Tuah Benua Kutai Timur. Data yang dibutuhkan yaitu berupa data jenis-jenis bagian pelayanan yang ada di PDAM Tirta Tuah Benua Kutai Timur serta data klasifikasi jenis keluhan pelanggan. Dimana data ini nantinya akan digunakan untuk mempermudah mengklasifikasikan keluhan pelanggan secara otomatis melalui inputan pengguna pada saat mengisi form keluhan pelanggan.

### 3.3 Analisis Kebutuhan

Analisis kebutuhan merupakan suatu tahap yang dilakukan dalam rangka mendapatkan daftar kebutuhan yang akan menyelesaikan masalah pengguna baik kebutuhan fungsional maupun non fungsional yang harus ada pada sebuah perangkat lunak yang akan dibangun. Kegiatan yang dilakukan untuk mendapatkan daftar kebutuhan yaitu dengan observasi langsung ke PDAM Tirta Tuah Benua Kutai Timur dan melakukan wawancara kepada karyawan.

Pada analisis kebutuhan fungsional akan dilakukan analisis untuk mendapatkan fungsi apa saja yang nantinya harus mampu ditangani oleh sistem. Daftar kebutuhan fungsional akan digambarkan dalam bentuk *use case* diagram yang akan ditunjukkan siapa saja aktor yang ada dan fungsi apa saja yang harus dapat dilakukan oleh aktor tersebut. Pada analisis kebutuhan non fungsional akan dilakukan analisis untuk mendapatkan apa saja yang harus dimiliki sebuah sistem guna memiliki kinerja yang baik pada saat sistem digunakan oleh user.

### 3.4 Perancangan Sistem

Tahap ini adalah tahap setelah analisis kebutuhan yang dilakukan dengan tujuan untuk memungkinkan aplikasi yang akan dibangun dapat mencakup seluruh daftar kebutuhan yang telah didapat pada saat proses analisis kebutuhan.

Beberapa tahapan yang dilakukan pada saat melakukan perancangan sistem antara lain:

#### 1. Perancangan Arsitektur Sistem

Pada langkah ini akan direpresentasikan relasi dari komponen yang menyusun aplikasi pelaporan keluhan. Komponen yang dimaksud antara lain perangkat mobile, *website*, Google Maps API, serta koneksi internet.

## 2. Perancangan Basis Data

Pada langkah ini akan di representasikan model dari penyimpanan data yang ada pada aplikasi pelaporan keluhan pelanggan. *Entity Relationship Diagram* (ERD) merupakan representasi dari model penyimpanan data pada aplikasi yang dibangun. Dimana pada ERD terdapat hubungan antar entitas yang dihubungkan dengan relasi tertentu kemudian akan dibuat struktur dari masing-masing tabel berdasarkan ERD pada *database* tersebut.

## 3. Perancangan Navigasi Antarmuka

Pada langkah ini akan dibuat rancangan antarmuka dari aplikasi pelaporan keluhan yang akan dibangun. Desain antarmuka dibuat menggunakan layout yang berisi berbagai komponen seperti *button*, *text field*, *label* yang terdapat pada halaman aplikasi tersebut. Kemudian akan dibuat perancangan navigasi yang berisi gambaran bagaimana alur tampilan pada halaman antarmuka aplikasi pelaporan keluhan yang telah dibuat.

### 3.5 Implementasi

Pada tahap implementasi, akan dilakukan dengan menggunakan perancangan aplikasi yang telah dilakukan sebelumnya sebagai acuan. *Software* yang digunakan untuk mengimplementasikan aplikasi antara lain Android Studio untuk aplikasi mobile serta Xampp serta sublime untuk aplikasi berbasis web.

Android Studio adalah *software* yang nantinya akan digunakan untuk mengimplementasikan kode program dengan bahasa pemrograman Java. Sedangkan sublime akan digunakan untuk mengimplementasikan kode program dengan bahasa pemrograman php serta akan di jalankan menggunakan *software* Xampp.

### 3.6 Pengujian dan Analisis

Pengujian merupakan tahap yang penting untuk dilakukan pada pembangunan perangkat lunak dengan kualitas yang tinggi, pengujian ini bertujuan untuk mengetahui apakah perangkat lunak yang dibangun telah berjalan sesuai dengan analisis kebutuhan serta perancangan yang telah dilakukan sebelumnya. Terdapat tiga pengujian yang dilakukan antara lain pengujian unit, pengujian validasi, serta pengujian *compatibility*.

Pengujian unit termasuk dalam pengujian *whitebox* yaitu menguji alur kode program yang telah diimplementasikan dengan cara membuat *flow graph* dari sebuah *method* yang ada pada kode program kemudian dihitung *cyclomatic complexity* dan akan menghasilkan jalur yang berbeda.

Pengujian validasi termasuk dalam pengujian *blackbox* yaitu dilakukan dengan cara menjalankan setiap fungsionalitas yang ada pada aplikasi pelaporan keluhan pelanggan. Kemudian dicari apakah terdapat perbedaan antara hasil yang didefinisikan pada perancangan dengan hasil pada aplikasi yang sudah jadi tanpa

berfokus pada alur kode program. Apabila aplikasi telah memenuhi kebutuhan fungsional yang sebelumnya telah dirancang maka akan menghasilkan nilai valid, namun pengujian akan memiliki nilai tidak valid apabila aplikasi tidak dapat memenuhi kebutuhan fungsional yang telah dirancang.

Pengujian *compatibility* pada penelitian ini dilakukan untuk menguji kompatibilitas aplikasi terhadap beberapa versi sistem operasi yang ada pada android. Pada pengujian ini aplikasi akan dijalankan pada 4 versi android yang berbeda yaitu Android 5.1 Lollipop, Android 6.0.1 Marshmallow, Android 7.0 Nougat, dan Android 8.0 Oreo.

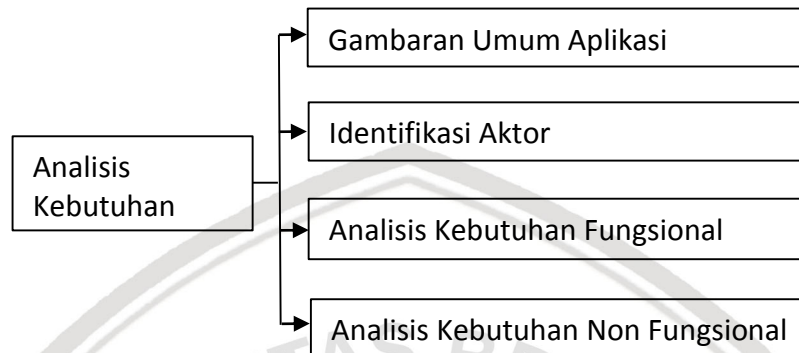
### 3.7 Kesimpulan dan Saran

Tahap terakhir yang dilakukan pada penelitian ini yaitu penarikan kesimpulan yang berdasarkan dengan rumusan masalah yang telah dibuat pada bab pendahuluan, serta berdasarkan pada tahap perancangan, implementasi, hasil pengujian dan analisis yang telah dilakukan dalam penelitian ini. Saran merupakan hal yang dibutuhkan oleh pengembang untuk dapat memperbaiki kesalahan-kesalahan yang ada serta masukan untuk adanya pengembangan selanjutnya.



## BAB 4 ANALISIS KEBUTUHAN

Bab ini berisi pembahasan tentang analisis dari aplikasi pelaporan keluhan pelanggan PDAM menggunakan fitur geotagging. Terdapat beberapa tahapan pada bab analisis kebutuhan yang akan digambarkan sebagai berikut:



**Gambar 4.1 Tahapan Analisis Kebutuhan**

Proses analisis kebutuhan dilaksanakan dalam rangka untuk mendapatkan daftar kebutuhan aplikasi pelaporan keluhan pelanggan PDAM baik kebutuhan fungsional maupun kebutuhan non fungsional. Untuk mendapatkan kebutuhan yang dapat memenuhi kebutuhan pengguna, tahapan yang dilakukan antara lain identifikasi aktor, analisis kebutuhan fungsional, analisis kebutuhan *non-fungsional*, menggambar *use case diagram*, *use case* skenario.

### 4.1 Gambaran Umum Aplikasi

Aplikasi pelaporan keluhan pelanggan PDAM menggunakan fitur *geotagging* berbasis android ini merupakan sebuah aplikasi yang dibuat untuk membantu masyarakat khususnya pelanggan PDAM Tirta Tuah Benua Kutai Timur sebagai pengguna utama aplikasi ini untuk melaporkan keluhannya terhadap berbagai permasalahan yang dihadapi terkait dengan PDAM Tirta Tuah Benua Kutai Timur. Selain itu pengguna juga dapat membaca informasi terkait pelaporan keluhan yang ada. Dengan adanya aplikasi ini pengguna dapat melaporkan berbagai keluhan seperti macetnya aliran air, kebocoran pipa, air yang bermasalah (keruh, bau), masalah pada tagihan air, serta keluhan lainnya. Pengguna akan lebih dipermudah untuk menunjukkan dimana lokasinya dengan adanya fitur *geotagging* yang tersedia pada aplikasi ini. Ketika pengguna mengunggah gambar keluhannya serta memberikan lokasi melalui google maps, maka pengguna tidak perlu memberikan lokasinya secara manual sehingga hasilnya akan lebih akurat.

### 4.2 Identifikasi Aktor

Pada tahapan identifikasi aktor ini akan dijelaskan siapa saja aktor yang akan melakukan interaksi dengan aplikasi pelaporan keluhan pelanggan PDAM serta apa saja yang dapat aktor tersebut lakukan. Aktor serta deskripsi dari peranan aktor tersebut dapat dilihat pada Tabel 4.1



**Tabel 4.1 Identifikasi Aktor**

Aktor	Deskripsi
Tamu	Pengguna aplikasi pelaporan keluhan pelanggan PDAM yang belum memiliki akun untuk mengakses aplikasi pelaporan keluhan.
User	Pengguna aplikasi pelaporan keluhan pelanggan PDAM yang dapat melaporkan keluhan yang berhubungan dengan PDAM
Admin	Pengguna website pelaporan keluhan pelanggan PDAM yang dapat memberi umpan balik terhadap laporan keluhan yang telah diberikan oleh user

### 4.3 Analisis Kebutuhan Fungsional

Pada tahap analisis kebutuhan fungsional akan dijabarkan fungsi apa saja yang terdapat pada aplikasi pelaporan keluhan pelanggan PDAM guna memenuhi kebutuhan pengguna. Daftar kebutuhan fungsional ini didapatkan dari observasi langsung yang dilakukan pada PDAM Tirta Tuah Benua Kutai Timur. Pada Tabel 4.3 dan Tabel 4.4 di definisikan nomor kebutuhan, nama fungsi serta deskripsi dari kebutuhan tersebut.

**Tabel 4.2 Kebutuhan Fungsional Aplikasi Untuk Tamu**

Nomor Kebutuhan	Nama fungsi	Deskripsi
SRS-01U-01	Melakukan Registrasi	Sistem harus dapat melakukan penyimpanan data akun baru ke dalam database
SRS-01U-02	Melakukan Login	Sistem harus dapat melakukan validasi data yang diinputkan user untuk masuk ke dalam akun yang telah terdaftar

**Tabel 4.3 Kebutuhan Fungsional Aplikasi Untuk User**

Nomor Kebutuhan	Nama fungsi	Deskripsi
SRS-01U-01	Melakukan Registrasi	Sistem harus dapat melakukan penyimpanan data akun baru ke dalam database
SRS-01U-02	Melakukan Login	Sistem harus dapat melakukan validasi data yang diinputkan user untuk masuk ke dalam akun yang telah terdaftar

Tabel 4.3 Kebutuhan Fungsional Aplikasi Untuk User (Lanjutan)

Nomor Kebutuhan	Nama fungsi	Deskripsi
SRS-01U-03	Melihat Home	Sistem harus dapat menampilkan halaman home yang berisi informasi terkait pelaporan keluhan untuk user
SRS-01U-04	Melaporkan Keluhan	Sistem harus dapat menyediakan form pelaporan keluhan dan melakukan penyimpanan data laporan user ke dalam database
SRS-01U-05	Mengambil gambar dari kamera	Sistem harus dapat mengambil gambar dari kamera
SRS-01U-06	Mengambil gambar dari <i>gallery</i>	Sistem harus dapat menampilkan gambar yang terdapat pada <i>smartphone</i> user dan mengambil foto yang dipilih user dari <i>gallery</i>
SRS-01U-07	Melihat Laporan Saya	Sistem harus dapat menampilkan daftar semua laporan yang telah dilaporkan oleh user
SRS-01U-08	Melihat detail Laporan	Sistem harus dapat menampilkan detail laporan dari laporan yang telah dipilih user
SRS-01U-09	Merubah Laporan	Sistem harus dapat melakukan perubahan data laporan user berdasarkan inputan user
SRS-01U-10	Menghapus Laporan	Sistem harus dapat menghapus laporan dari daftar laporan
SRS-01U-11	Melakukan Logout	Sistem harus dapat melakukan penghapusan session agar user dapat keluar dari akunnya

Tabel 4.4 Kebutuhan Fungsional Admin

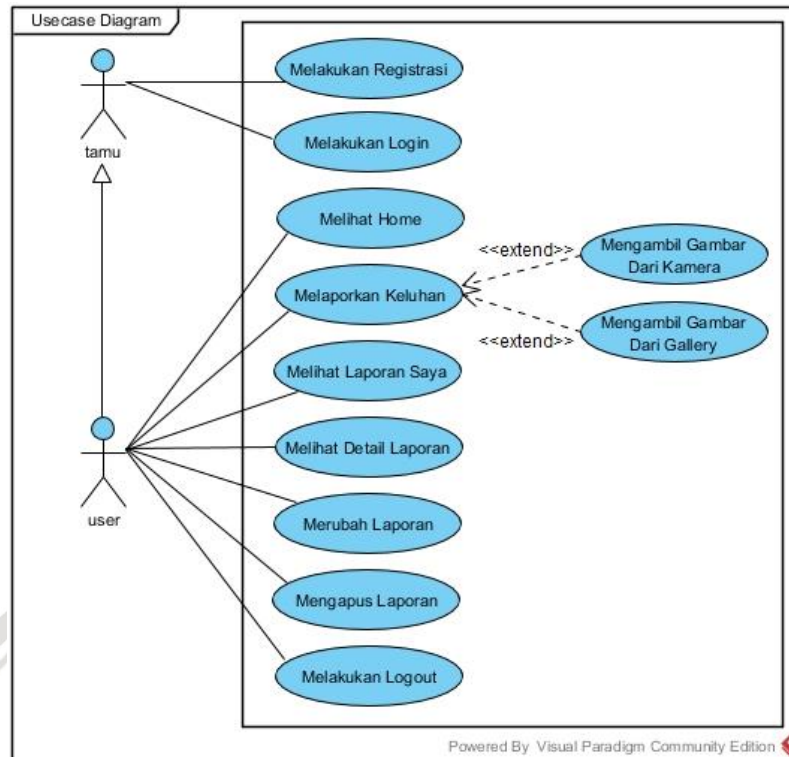
Nomor Kebutuhan	Nama fungsi	Deskripsi
SRS-01A-01	Melakukan Login	Sistem harus dapat melakukan validasi data yang diinputkan admin untuk masuk ke dalam akun yang telah terdaftar
SRS-01A-02	Melihat Detail Laporan	Sistem harus dapat menampilkan detail laporan dari laporan user yang telah dipilih oleh admin

Tabel 4.4 Kebutuhan Fungsional Admin (Lanjutan)

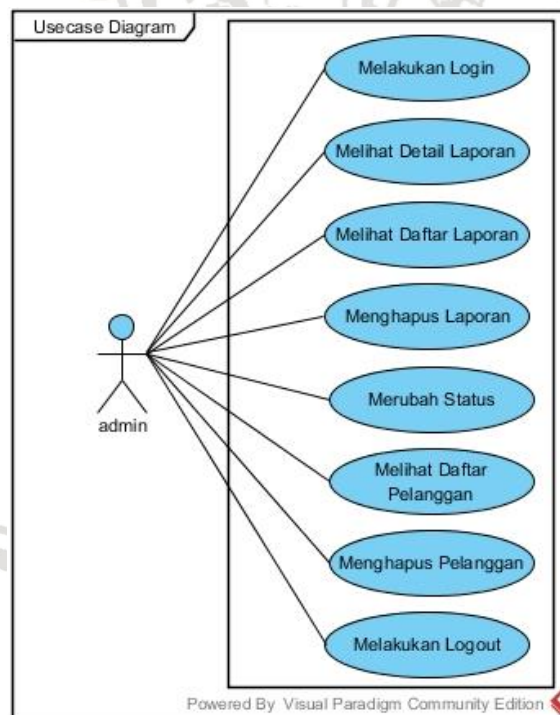
Nomor Kebutuhan	Nama fungsi	Deskripsi
SRS-01A-03	Melihat Daftar Laporan	Sistem harus dapat menampilkan daftar semua laporan yang telah dilaporkan oleh user untuk dapat dilihat oleh admin
SRS-01A-04	Menghapus laporan	Sistem harus dapat menghapus laporan user yang dipilih admin dari daftar laporan
SRS-01A-05	Merubah Status Laporan	Sistem harus dapat melakukan perubahan data status laporan berdasarkan inputan yang diberikan admin
SRS-01A-06	Melihat Daftar Pelanggan	Sistem harus dapat menampilkan daftar semua pelanggan yang telah terdaftar untuk dapat dilihat oleh admin
SRS-01A-07	Menghapus Pelanggan	Sistem harus dapat menghapus pelanggan sebagai user yang dipilih admin dari daftar pelanggan
SRS-01A-08	Melakukan Logout	Sistem harus dapat melakukan penghapusan session agar admin dapat keluar dari akunnya

#### 4.3.1 Use Case Diagram

Pada Gambar 4.2 dan Gambar 4.3 akan di gambarkan *use case* diagram dari aplikasi pelaporan keluhan pelanggan PDAM, dimana terdapat dua *use case* diagram yaitu *use case* diagram untuk aplikasi *mobile* dan *use case* diagram untuk *website* admin. Pada *use case* diagram aplikasi *mobile* yang ditampilkan pada Gambar 4.2 terdapat 2 aktor dengan nama tamu dan user, dimana pada aktor tamu memiliki hubungan asosiasi dengan 2 *use case* yaitu registrasi dan login. Aktor user memiliki hubungan generalisasi terhadap aktor tamu, yang berarti fungsi registrasi dan login yang dimiliki aktor tamu akan diturunkan ke user sehingga user juga dapat melakukan fungsi tersebut. Selain itu user memiliki hubungan asosiasi dengan 7 *use case* yaitu melihat home, melaporkan keluhan, melihat daftar laporan, melihat detail laporan, merubah status, menghapus laporan, dan melakukan *logout* serta terdapat 2 *use case* tambahan yaitu mengambil gambar dari kamera dan mengambil gambar dari *gallery* yang memiliki hubungan *extend* dengan *use case* melaporkan keluhan.



Gambar 4.2 Use Case Diagram User



Gambar 4.3 Use Case Diagram Website Admin

Pada *use case diagram website admin* yang digambarkan pada Gambar 4.3 terdapat 1 aktor dengan nama admin yang memiliki hubungan asosiasi dengan 8 *use case*.

### 4.3.2 Use Case Skenario

Setelah *use case* diagram selesai dibuat, selanjutnya dijelaskan *use case* skenario berdasarkan *use case* diagram dengan mendefinisikan skenario dari setiap *use case* yang ada. Pada *use case* skenario ini berisi tujuan (*objective*) dari *use case*, aktor yang berinteraksi dengan *use case*, kondisi awal (*pre-condition*) yang harus dipenuhi, alur kerja (*main flow*) dari *use case*, alur kerja alternatif (*alternative flow*) yang akan dijalankan ketika *main flow* tidak terpenuhi, serta kondisi akhir (*post-condition*) yang harus dicapai. *Use case* skenario dari aplikasi pelaporan keluhan pelanggan PDAM akan di jelaskan pada Tabel 4.5 sampai Tabel 4.23.

**Tabel 4.5 Use Case Skenario Melakukan Registrasi**

<i>Objective</i>	Aktor dapat mendaftar untuk mendapatkan akun Aplikasi Pelaporan Keluhan	
Aktor	Tamu, User	
<i>Pre-Condition</i>	Aktor telah membuka aplikasi pelaporan keluhan	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan fungsi registrasi 3. Aktor mengisi data diri pada form registrasi (Nama Lengkap, no pelanggan, <i>username</i> , <i>email</i> , <i>password</i> , no hp) 4. Aktor menekan tombol register	2. Sistem menampilkan form registrasi 5. Sistem mengirimkan data registrasi ke dalam database untuk disimpan dan menampilkan halaman login
<i>Alternative Flow</i>	1. Apabila data diri tidak terisi secara lengkap, maka sistem akan menampilkan pesan kesalahan "data diri anda belum lengkap!"	
<i>Post Condition</i>	Aktor berhasil mendapatkan akun Aplikasi Pelaporan Keluhan	

**Tabel 4.6 Use Case Skenario Melakukan Login User**

<i>Objective</i>	Aktor dapat masuk ke dalam akun Aplikasi Pelaporan Keluhan	
Aktor	Tamu, User	
<i>Pre-Condition</i>	Aktor telah mengakses Aplikasi Pelaporan Keluhan	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan fungsi login 3. Aktor mengisi <i>username</i> serta <i>password</i> yang terdaftar	2. Sistem menampilkan form login



	4. Aktor menekan tombol login	5. Sistem menampilkan halaman "home"
<i>Alternative Flow</i>	1. Apabila kombinasi username dan password tidak terisi, salah atau tidak ada dalam daftar basis data sistem, maka sistem akan menampilkan pesan kesalahan "username atau password anda salah"	
<i>Post Condition</i>	Aktor berhasil masuk ke dalam akun Aplikasi Pelaporan Keluhan	

**Tabel 4.7 Use Case Skenario Melakukan Login Admin**

<i>Objective</i>	Aktor dapat masuk ke dalam akun <i>Website</i> Pelaporan Keluhan	
<i>Aktor</i>	Admin	
<i>Pre-Condition</i>	Aktor telah mengakses <i>Website</i> Pelaporan Keluhan	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor mengisi <i>username</i> serta <i>password</i> yang terdaftar 2. Aktor menekan tombol login	3. Sistem menampilkan halaman "daftar laporan"
<i>Alternative Flow</i>	1. Apabila kombinasi <i>username</i> dan <i>password</i> salah atau tidak terdaftar dalam sistem, maka sistem akan menampilkan pesan kesalahan 2. Apabila <i>username</i> atau <i>password</i> kosong maka sistem akan menampilkan pesan kesalahan " <i>please fill out this field</i> "	
<i>Post Condition</i>	Aktor berhasil masuk ke dalam akun <i>Website</i> Pelaporan Keluhan	

**Tabel 4.8 Use Case Skenario Melihat Home User**

<i>Objective</i>	Aktor dapat melihat halaman <i>home</i> yang berisi informasi	
<i>Aktor</i>	User	
<i>Pre-Condition</i>	Aktor telah masuk kedalam akun Aplikasi Pelaporan Keluhan	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tab <i>Home</i>	2. Sistem menampilkan halaman <i>home</i> yang berisi informasi terkait pelaporan keluhan
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Halaman <i>home</i> telah berhasil ditampilkan kepada aktor	

**Tabel 4.9 Use Case Skenario Melaporkan Keluhan oleh user**

<i>Objective</i>	Aktor dapat melaporkan keluhan kepada PDAM
------------------	--

Aktor	User	
Pre-Condition	Aktor telah masuk ke dalam akun Aplikasi Pelaporan Keluhan	
Main Flow	Aktor	Sistem
	<ol style="list-style-type: none"> <li>1. Aktor menekan tab Lapor</li> <li>3. Aktor mengambil gambar untuk dilaporkan</li> <li>5. Aktor mengisi semua field pada form pelaporan keluhan (Deskripsi, Bagian Pelaporan)</li> <li>6. Aktor menekan tombol "submit"</li> </ol>	<ol style="list-style-type: none"> <li>2. Sistem menampilkan halaman lapor keluhan</li> <li>4. Sistem mengambil data latitude dan longitude dari gambar dan menampilkan halaman form pelaporan keluhan</li> <li>7. Sistem menyimpan data laporan keluhan kedalam <i>database</i> dan menampilkan halaman home dan notifikasi laporan berhasil dikirim</li> </ol>
Alternative Flow	1. Apabila data laporan tidak diisi dengan lengkap, maka sistem akan menampilkan pesan kesalahan "Deskripsi tidak boleh kosong!"	
Post Condition	Data laporan keluhan aktor telah berhasil tersimpan kedalam <i>database</i> sistem	

**Tabel 4.10 Use Case Skenario Mengambil Gambar dari Kamera**

Objective	Aktor dapat mengambil gambar menggunakan kamera	
Aktor	User	
Pre-Condition	Aktor telah menekan tab Laporkan Keluhan	
Main Flow	Aktor	Sistem
	<ol style="list-style-type: none"> <li>1. Aktor menekan tab kamera</li> <li>3. Aktor menekan ikon <i>capture</i> untuk mengambil gambar</li> </ol>	<ol style="list-style-type: none"> <li>2. Sistem mengakses kamera <i>smartphone</i></li> </ol>
Alternative Flow	Apabila lokasi aktor tidak terdeteksi oleh sistem maka muncul pesan kesalahan "Foto Tidak Memiliki Koordinat! Silahkan cek GPS anda atau pilih foto dari galeri"	
Post Condition	Sistem telah berhasil mengambil foto melalui kamera	

**Tabel 4.11 Use Case Skenario Mengambil Gambar dari Gallery**

<i>Objective</i>	Aktor dapat mengambil foto dari gallery	
<i>Aktor</i>	User	
<i>Pre-Condition</i>	Aktor telah menekan tab Laporkan Keluhan	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tab <i>gallery</i>	2. Sistem menampilkan gambar yang tersimpan dalam <i>smartphone</i> aktor
	3. Aktor memilih gambar yang ingin dilaporkan	
<i>Alternative Flow</i>	Apabila foto yang dipilih aktor tidak memiliki informasi lokasi, maka akan menampilkan pesan kesalahan “Foto Tidak Memiliki Koordinat! Silahkan cek GPS anda atau pilih foto dari galeri”	
<i>Post Condition</i>	Sistem telah berhasil mengambil gambar dari gallery device	

**Tabel 4.12 Use Case Skenario Melihat Laporan Saya**

<i>Objective</i>	Aktor dapat melihat daftar laporan yang telah dilaporkan	
<i>Aktor</i>	User	
<i>Pre-Condition</i>	Aktor telah masuk kedalam akun Aplikasi Pelaporan Keluhan	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tab Laporan Saya	2. Sistem menampilkan daftar laporan
<i>Alternative Flow</i>	1. Apabila aktor belum melakukan laporan maka sistem akan menampilkan pesan kesalahan “Belum Ada Laporan!”	
<i>Post Condition</i>	Daftar laporan aktor berhasil ditampilkan oleh sistem	

**Tabel 4.13 Use Case Skenario Melihat Detail Laporan oleh User**

<i>Objective</i>	Aktor dapat melihat detail laporan yang telah dilaporkan	
<i>Aktor</i>	User	
<i>Pre-Condition</i>	Aktor telah berada di halaman Laporan Saya	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan gambar laporan yang diinginkan	2. Sistem menampilkan pilihan menu
	3. Aktor memilih menu “detail laporan”	4. Sistem menampilkan detail laporan
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Detail laporan berhasil ditampilkan oleh sistem	

**Tabel 4.14 Use Case Skenario Melihat Detail Laporan oleh Admin**

<i>Objective</i>	Aktor dapat melihat detail laporan yang telah dilaporkan	
<i>Aktor</i>	Admin	
<i>Pre-Condition</i>	Aktor telah berada di halaman daftar laporan	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol “Lihat” pada laporan yang diinginkan	2. Sistem menampilkan detail laporan yang dipilih
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Detail laporan user telah berhasil ditampilkan oleh sistem	

**Tabel 4.15 Use Case Skenario Merubah Laporan**

<i>Objective</i>	Aktor dapat merubah data laporan yang telah dilaporkan	
<i>Aktor</i>	User	
<i>Pre-Condition</i>	Aktor telah masuk kedalam halaman laporan saya	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan gambar laporan yang ingin dirubah 3. Aktor menekan menu ubah 5. Aktor merubah data laporan 6. Aktor menekan tombol submit	2. Sistem menampilkan pilihan menu 4. Sistem menampilkan form ubah laporan 7. Sistem menyimpan data laporan baru di database dan menampilkan halaman laporan saya
<i>Alternative Flow</i>	Apabila data laporan tidak diisi dengan lengkap, maka sistem akan menampilkan pesan kesalahan “Deskripsi tidak boleh kosong!”	
<i>Post Condition</i>	Data laporan telah berhasil terganti dengan data yang baru	

**Tabel 4.16 Use Case Skenario Menghapus Laporan User**

<i>Objective</i>	Aktor dapat menghapus data laporan yang telah dilaporkan	
<i>Aktor</i>	User	
<i>Pre-Condition</i>	Aktor telah masuk kedalam halaman laporan saya	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan gambar laporan yang ingin dihapus 3. Aktor menekan menu hapus	2. Sistem menampilkan pilihan menu 4. Sistem menghapus data laporan dari database dan

		menampilkan halaman laporan saya
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Data laporan telah berhasil terhapus	

**Tabel 4.17 Use Case Skenario Melihat Daftar Laporan Admin**

<i>Objective</i>	Aktor dapat melihat daftar laporan yang telah dilaporkan oleh user	
<i>Aktor</i>	Admin	
<i>Pre-Condition</i>	Aktor telah masuk kedalam akun Website Pelaporan Keluhan	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tab Daftar Laporan	2. Sistem menampilkan daftar laporan user
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Daftar laporan user berhasil ditampilkan oleh sistem kepada aktor	

**Tabel 4.18 Use Case Skenario Menghapus Laporan oleh Admin**

<i>Objective</i>	Aktor dapat menghapus Laporan User	
<i>Aktor</i>	Admin	
<i>Pre-Condition</i>	Aktor telah berada di halaman daftar laporan	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol hapus pada laporan yang ingin dihapus	2. Sistem menghapus data laporan dari <i>database</i> dan menampilkan halaman daftar laporan
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Data laporan keluhan pelanggan telah berhasil dihapus dari <i>database</i>	

**Tabel 4.19 Use Case Skenario Merubah Status Laporan**

<i>Objective</i>	Aktor dapat merubah status data laporan yang telah dilaporkan oleh user	
<i>Aktor</i>	Admin	
<i>Pre-Condition</i>	Aktor telah berada di halaman daftar laporan	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol edit	2. Sistem menampilkan form laporan keluhan untuk diedit
	3. Aktor memilih status laporan yang baru	



	4. Aktor menekan tombol submit	5. Sistem menyimpan data laporan keluhan yang telah diedit ke dalam <i>database</i> dan menampilkan halaman daftar laporan
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Status data laporan keluhan telah berhasil dirubah dan tersimpan di dalam <i>database</i>	

**Tabel 4.20 Use Case Skenario Melihat Daftar Pelanggan**

<i>Objective</i>	Aktor dapat melihat daftar pelanggan yang telah terdaftar	
Aktor	Admin	
<i>Pre-Condition</i>	Aktor telah masuk kedalam akun Website Pelaporan Keluhan	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tab Daftar Pelanggan	2. Sistem menampilkan daftar pelanggan yang terdaftar
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Daftar pelanggan berhasil ditampilkan oleh sistem kepada aktor	

**Tabel 4.21 Use Case Skenario Menghapus Pelanggan**

<i>Objective</i>	Aktor dapat menghapus pelanggan	
Aktor	Admin	
<i>Pre-Condition</i>	Aktor telah berada di halaman daftar pelanggan	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol hapus pada pelanggan yang ingin dihapus	2. Sistem menghapus data pelanggan dari <i>database</i> dan menampilkan halaman daftar pelanggan
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Data pelanggan telah berhasil dihapus dari <i>database</i>	

**Tabel 4.22 Use Case Skenario Melakukan Logout User**

<i>Objective</i>	Aktor dapat keluar dari akun aplikasi pelaporan keluhan	
Aktor	User	
<i>Pre-Condition</i>	Aktor telah masuk kedalam akun sistem	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol logout	2. Sistem mengeluarkan aktor dari akun sistem dan menampilkan halaman login

<i>Alternative Flow</i>	-
<i>Post Condition</i>	Aktor berhasil keluar dari akun dan session dihapus

**Tabel 4.23 Use Case Skenario Melakukan Logout Admin**

<i>Objective</i>	Aktor dapat keluar dari akun website pelaporan keluhan	
<i>Aktor</i>	Admin	
<i>Pre-Condition</i>	Aktor telah masuk kedalam akun sistem	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor mengeklik tombol logout	2. Sistem mengeluarkan aktor dari akun sistem dan menampilkan halaman login
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Aktor berhasil keluar dari akun dan session dihapus	

#### 4.4 Analisis Kebutuhan Non Fungsional

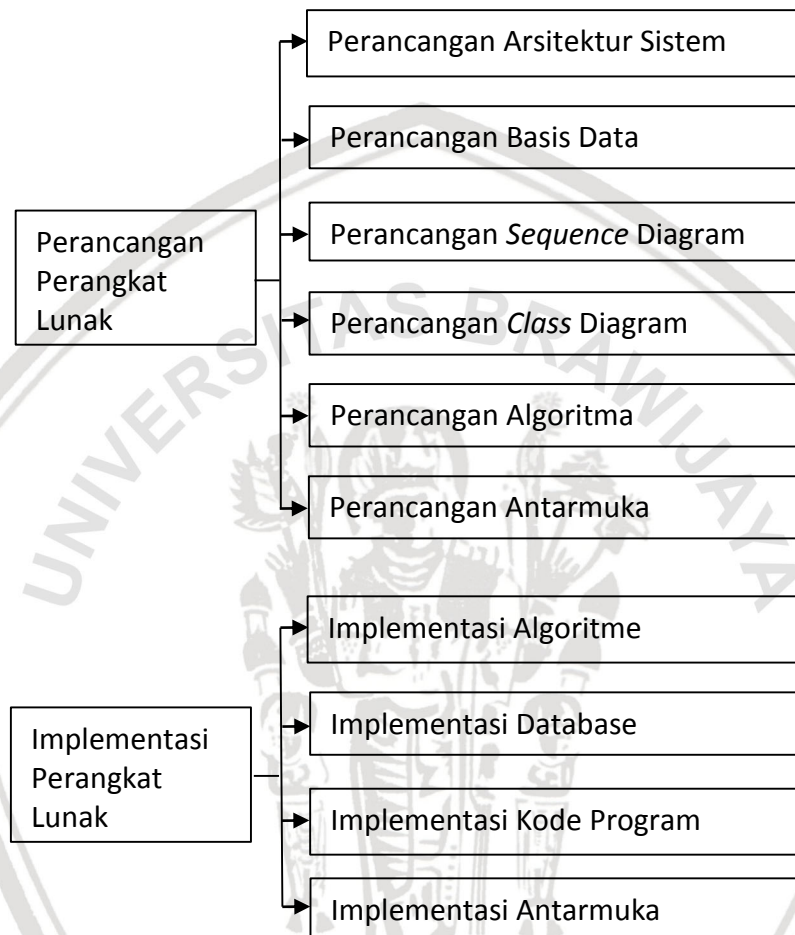
Analisis kebutuhan non-fungsional dilakukan dalam rangka untuk menentukan kualitas dari keseluruhan perangkat lunak yang dibangun. Kebutuhan fungsional dari penelitian ini yaitu Usability. Nama kebutuhan non fungsional serta deskripsinya akan dijabarkan pada Tabel 4.24

**Tabel 4.24 Kebutuhan Non-Fungsional**

Nomor Kebutuhan	Parameter Kebutuhan	Deskripsi
SRS-02-01	<i>Compatibility</i>	Aplikasi dapat dijalankan pada berbagai perangkat mobile dengan versi OS yang berbeda yaitu 5.1 Lollipop, 6.0.1 Marshmallow, 7.0 Nougat, dan 8.0 Oreo.

## BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini menjelaskan tentang perancangan serta implementasi dari aplikasi pelaporan keluhan pelanggan PDAM menggunakan fitur geotagging. Terdapat beberapa tahapan pada bab perancangan dan implementasi yang akan digambarkan sebagai berikut:



Gambar 5.1 Tahapan Perancangan Perangkat Lunak

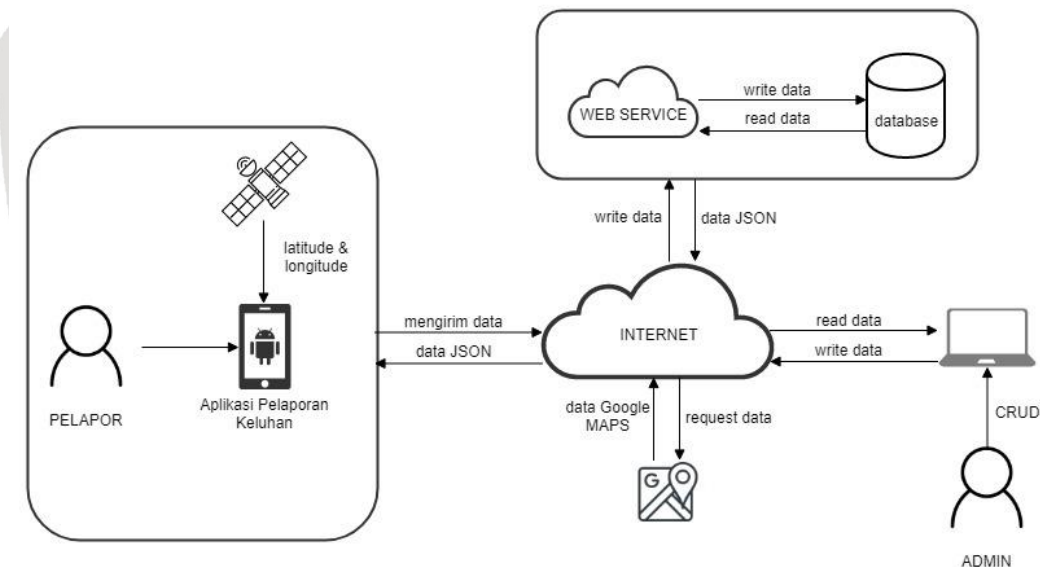
### 5.1 Perancangan Perangkat Lunak

Pada tahapan perancangan perangkat lunak terdapat beberapa tahapan, antara lain yaitu perancangan arsitektur sistem yang akan menjelaskan tentang arsitektur dari sistem yang dibangun yang berisi komponen-komponen yang terlibat pada saat sistem dijalankan, perancangan basis data yang akan digambarkan dengan *Entity Relationship Diagram* (ERD) yang berisi entitas-entitas serta hubungan antar entitas tersebut dan yang terakhir perancangan antarmuka dari setiap halaman yang ada pada aplikasi pelaporan keluhan pelanggan PDAM.

### 5.1.1 Perancangan Arsitektur Sistem

Pada Gambar 5.2 dijelaskan perancangan arsitektur pada aplikasi pelaporan keluhan pelanggan PDAM. Pada sisi *client* terdapat user yang dapat mengakses aplikasi *mobile*, dimana user merupakan pelanggan yang ingin melaporkan keluhannya. Ketika akan melaporkan keluhannya pengguna harus mengisi form dan mengupload gambar, kemudian aplikasi akan melakukan proses *geotagging* dan mencari tahu lokasi pengguna dengan menggunakan teknologi *global positioning system* (GPS). GPS akan memberikan informasi berupa latitude dan longitude kepada aplikasi mobile. Setelah itu data akan dikirimkan ke database untuk disimpan melalui *web service* dengan menggunakan jaringan internet, dimana *database* yang digunakan yaitu MySQL.

Selain user, terdapat admin yang dapat mengakses website pelaporan keluhan pelanggan PDAM. Admin dapat melihat data laporan pelanggan yang melalui *web service* yang mengambil data pada *database* dengan menggunakan format JSON, serta peta yang menampilkan lokasi pelanggan yang telah melapor dalam bentuk *marker* dengan melakukan *request* ke Google Maps API. Gambar Arsitektur sistem dapat dilihat pada Gambar 5.2.



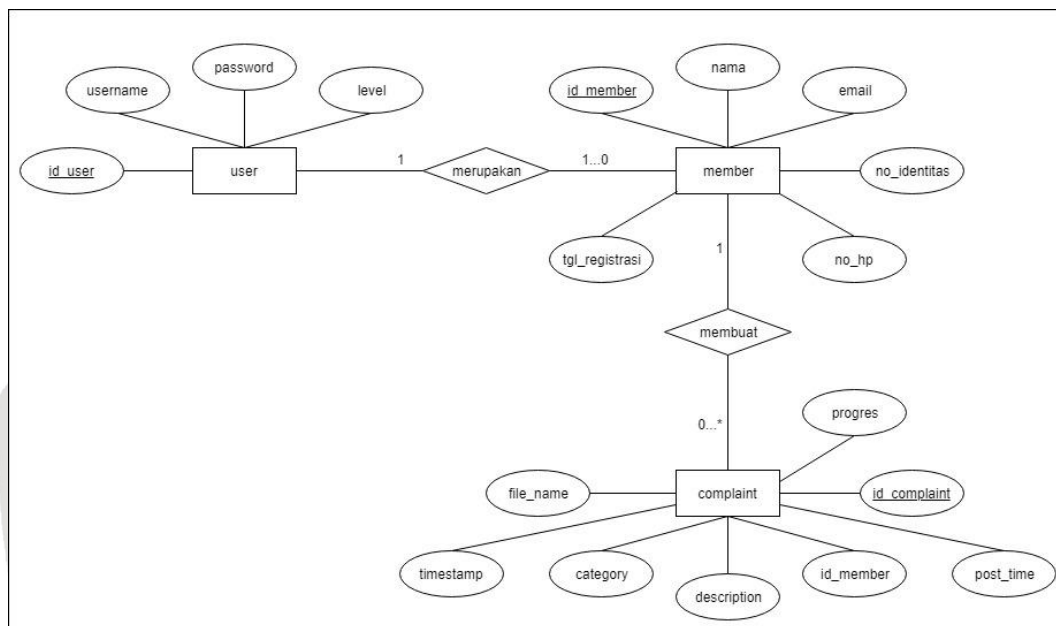
**Gambar 5.2 Arsitektur Sistem dari Aplikasi Pelaporan Keluhan Pelanggan PDAM**

### 5.1.2 Perancangan Basis Data

Pada tahap ini akan dijelaskan bagaimana rancangan basis data untuk menyimpan berbagai data yang terkait dengan aplikasi pelaporan keluhan pelanggan PDAM. Basis data yang digunakan yaitu basis data *online storage* sehingga ketika melakukan pemrosesan data diperlukan adanya koneksi internet. Perancangan basis data direpresentasikan dengan *Entity Relationship Diagram* (ERD), dimana di dalam ERD tersebut berisi entitas-entitas beserta relasinya yang dilengkapi dengan atribut. Dalam perancangan basis data pada aplikasi pelaporan

keluhan pelanggan PDAM, terdapat 3 entitas yaitu user, member dan complaint yang masing-masing entitas memiliki atributnya.

Entitas user adalah entitas yang menyimpan data dari semua pengguna yang dapat mengakses sistem, baik tamu, user maupun admin. Entitas member merupakan entitas dari pengguna yang menyimpan data berupa data diri pengguna secara lengkap. Entitas complaint merupakan entitas yang menyimpan data yang berkaitan dengan keluhan yang dikirim oleh pengguna. Pada Gambar 5.3 akan direpresentasikan *Entity Relationship Diagram* dari aplikasi pelaporan keluhan pelanggan PDAM.



**Gambar 5.3 Entity Relationship Diagram**

Berdasarkan *Entity Relationship Diagram* pada Gambar 5.3 terdapat 3 entitas dan 2 relasi, dimana entitas tersebut akan direpresentasikan dalam bentuk tabel pada basis data yang akan dibuat. Sehingga dalam basis data yang akan dibuat terdapat 3 tabel dengan nama User, Member, dan Complaint. Pada Tabel 5.1 di definisikan rancangan struktur table user yang merupakan gambaran dari entitas user. Tabel user berisi data-data yang terkait dengan pengguna yang terdaftar baik user maupun admin yang digunakan untuk dapat mengakses sistem. Pada tabel ini terdapat 4 field yaitu *id\_user*, *username*, *password*, serta *level*.

**Tabel 5.1 Rancangan Struktur Tabel User**

No.	Nama Field	Tipe Data	Keterangan
1.	<i>id_user</i>	<i>Integer</i>	Nomor identitas yang dimiliki oleh pengguna
2.	<i>username</i>	<i>Varchar</i>	<i>Username</i> yang dimiliki oleh pengguna untuk dapat masuk kedalam akun sistem
3.	<i>password</i>	<i>Varchar</i>	<i>Password</i> yang dimiliki oleh pengguna untuk dapat masuk kedalam akun sistem



4.	<i>level</i>	<i>Varchar</i>	Level yang dimiliki pengguna untuk dapat membedakan user dan admin
----	--------------	----------------	--

Tabel *Member* merupakan tabel yang berisi data-data yang diisikan oleh pengguna pada saat melakukan registrasi ke dalam sistem. Tabel ini merupakan representasi dari entitas member yang memiliki 6 atribut. Rancangan struktur tabel member secara lengkap didefinisikan pada Tabel 5.2

**Tabel 5.2 Rancangan Struktur Tabel Member**

No.	Nama <i>Field</i>	Tipe Data	Keterangan
1.	<i>id_member</i>	<i>Integer</i>	Nomor identitas yang dimiliki oleh pelanggan sebagai user
2.	<i>nama</i>	<i>Varchar</i>	Nama lengkap pelanggan
3.	<i>no_identitas</i>	<i>Varchar</i>	Nomor pelanggan yang dimiliki oleh pelanggan sebagai user
4.	<i>email</i>	<i>Varchar</i>	Alamat email yang dimiliki oleh pelanggan
5.	<i>no_hp</i>	<i>Varchar</i>	No Hp yang dimiliki oleh pelanggan
6.	<i>tgl_registrasi</i>	<i>Datetime</i>	Tanggal pada saat pelanggan melakukan registrasi

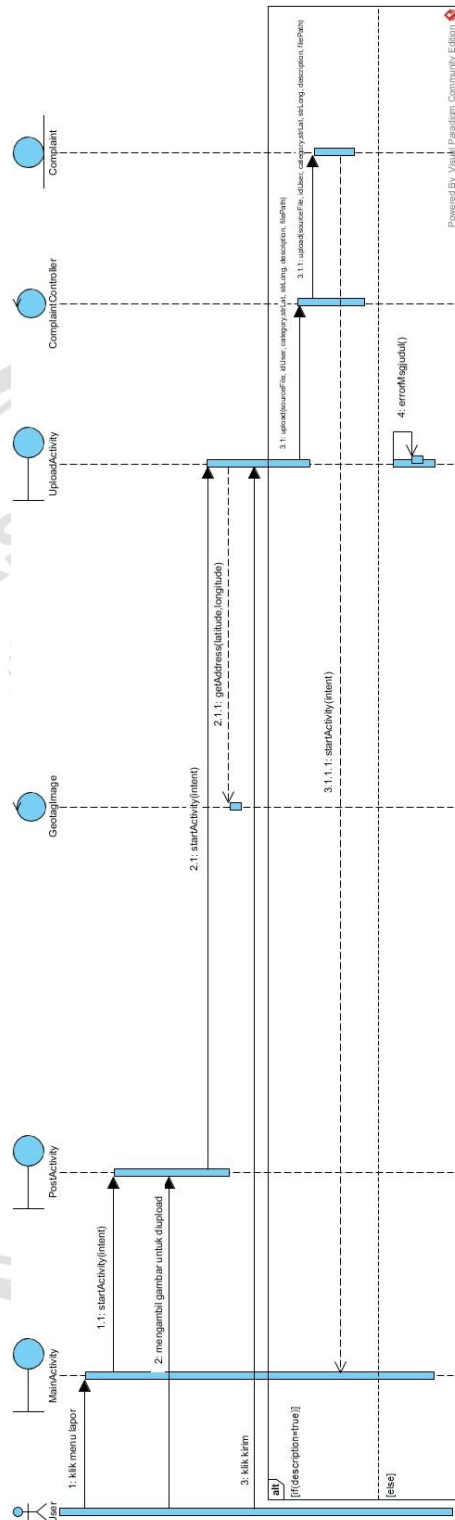
Tabel *Complaint* merupakan tabel yang berisi data-data terkait laporan yang dikirimkan oleh user. Tabel ini merupakan representasi dari entitas laporan yang memiliki 8 atribut. Rancangan struktur tabel laporan secara lengkap didefinisikan pada Tabel 5.3

**Tabel 5.3 Rancangan Struktur Tabel Complaint**

No.	Nama <i>Field</i>	Tipe Data	Keterangan
1.	<i>id_laporan</i>	<i>Integer</i>	Nomor identitas dari laporan yang dibuat
2.	<i>id_member</i>	<i>Integer</i>	Nomor identitas yang dimiliki oleh pelanggan sebagai user
3.	<i>category</i>	<i>Varchar</i>	Bagian pelayanan laporan yang dipilih oleh pelanggan sebagai user
4.	<i>description</i>	<i>Text</i>	Deskripsi dari laporan yang dibuat
5.	<i>file_name</i>	<i>Varchar</i>	Nama dari file gambar yang telah di upload
6.	<i>progres</i>	<i>Varchar</i>	Progress dari pengerjaan laporan
7.	<i>post_time</i>	<i>Datetime</i>	Tanggal pada saat melakukan <i>post</i> keluhan
8.	<i>timestamp</i>	<i>Timestamp</i>	Waktu pada saat melakukan <i>post</i> keluhan

### 5.1.3 Perancangan *Sequence Diagram*

#### 5.1.3.1 Fungsi Melaporkan Keluhan oleh user

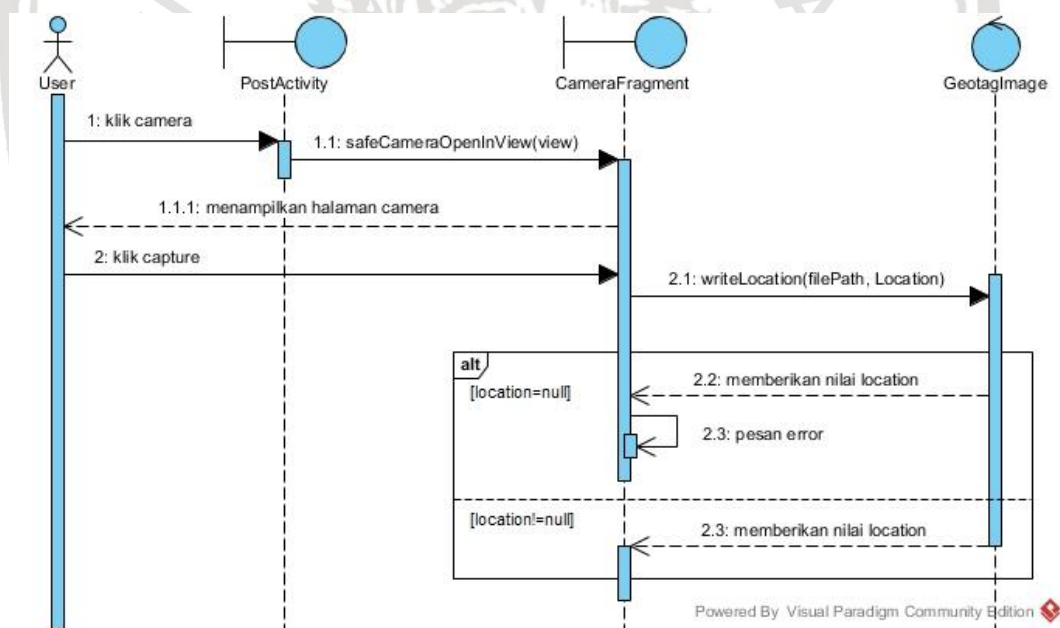


Gambar 5.4 *Sequence Diagram* Melaporkan Keluhan

Pada Gambar 5.4 menunjukkan pertukaran pesan antar objek yang digambarkan dalam bentuk lifeline yang berurutan berdasarkan urutan waktu. Dalam *sequence* diagram Melaporkan Keluhan terdapat 1 aktor bernama user dan 8 lifeline antara lain yaitu 3 boundary dengan nama MainActivity, PostActivity, dan UploadActivity. Kemudian terdapat 2 controller yang menghubungkan *boundary* dengan *entity* dengan nama GeotagImage dan ComplaintController. Dan yang terakhir terdapat 1 *entity* dengan nama Complaint.

Fungsi Melaporkan Keluhan ini dimulai ketika User sebagai Aktor menekan menu lapor yang akan menjalankan boundary PostActivity dengan memanggil method `startActivity(intent)` untuk mengambil gambar baik melalui kamera ataupun gallery. Kemudian user mengisi judul serta deskripsi dari laporan yang akan dilaporkan lalu menekan tombol kirim, pada kondisi ini terdapat 2 alternatif yang ada, yaitu ketika deskripsi laporan terisi, serta deskripsi laporan kosong. Ketika deskripsi laporan terisi, maka UploadActivity akan memanggil *method* `upload()` dengan parameter berupa data laporan yang ada pada *controller* ComplaintController yang akan langsung diteruskan pada entity Controller untuk disimpan kedalam *database* dengan memanggil *method* `upload()` dengan isi parameter yang sama.

#### 5.1.3.2 Fungsi Mengambil Gambar dari Kamera oleh user



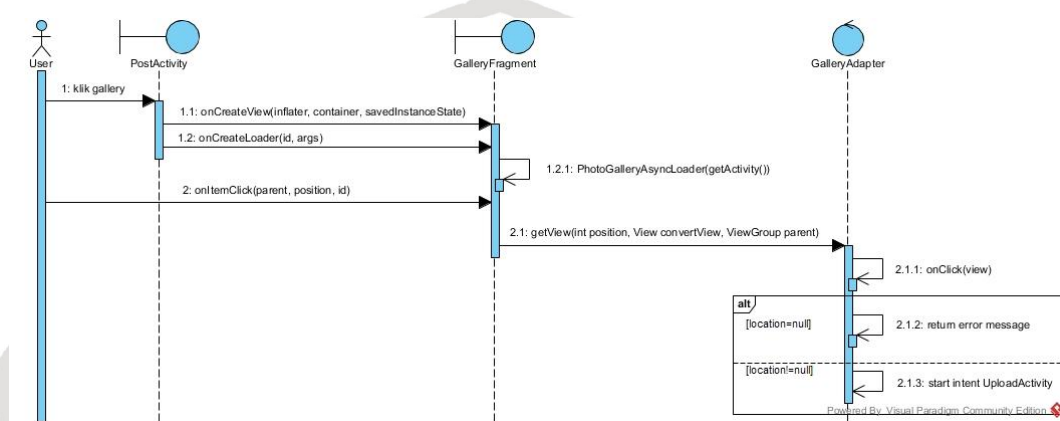
Gambar 5.5 Sequence Diagram Mengambil Gambar dari Kamera

Pada Gambar 5.5 digambarkan *sequence* diagram fungsi mengambil gambar dari kamera. Terdapat 1 aktor bernama user dan 3 *lifeline* antara lain yaitu 2 *boundary* dengan nama PostActivity dan CameraFragment serta 1 *controller* dengan nama GeotagImage.

Pada saat user menekan menu kamera maka akan langsung mengirim pesan ke *boundary* PostActivity untuk membuka *camera* pada *boundary* CameraFragment dengan memanggil *method* `saveCameraOpenInView(view)`.

Setelah *camera* terbuka dan user mengambil gambar, maka aplikasi akan langsung melakukan proses *geotagging* dengan memanggil *method* `writeLocation(filePath, Location)` pada *controller* `GeotagImage`. Dari proses *geotagging* ini akan menghasilkan nilai *location*. Kemudian terdapat 2 alternatif, ketika *location* bernilai *null* yang berarti lokasi tidak ditemukan maka akan menampilkan pesan error pada *boundary* `CameraFragment`. Kemudian pada alternatif kedua yaitu ketika *location* tidak bernilai *null* atau lokasi ditemukan maka akan dilanjutkan dengan menjalankan form pelaporan keluhan dimana prosesnya akan digambarkan pada *sequence diagram* melaporkan keluhan.

### 5.1.3.3 Fungsi Mengambil Foto dari Gallery



Gambar 5.6 Sequence Diagram Mengambil Foto dari Gallery

Pada Gambar 5.6 digambarkan *sequence diagram* fungsi mengambil foto dari *gallery*. Terdapat 1 aktor bernama user dan 3 *lifeline* antara lain yaitu 2 *boundary* dengan nama `PostActivity` dan `GalleryFragment` serta terdapat 1 *controller* yang bernama `GalleryAdapter`.

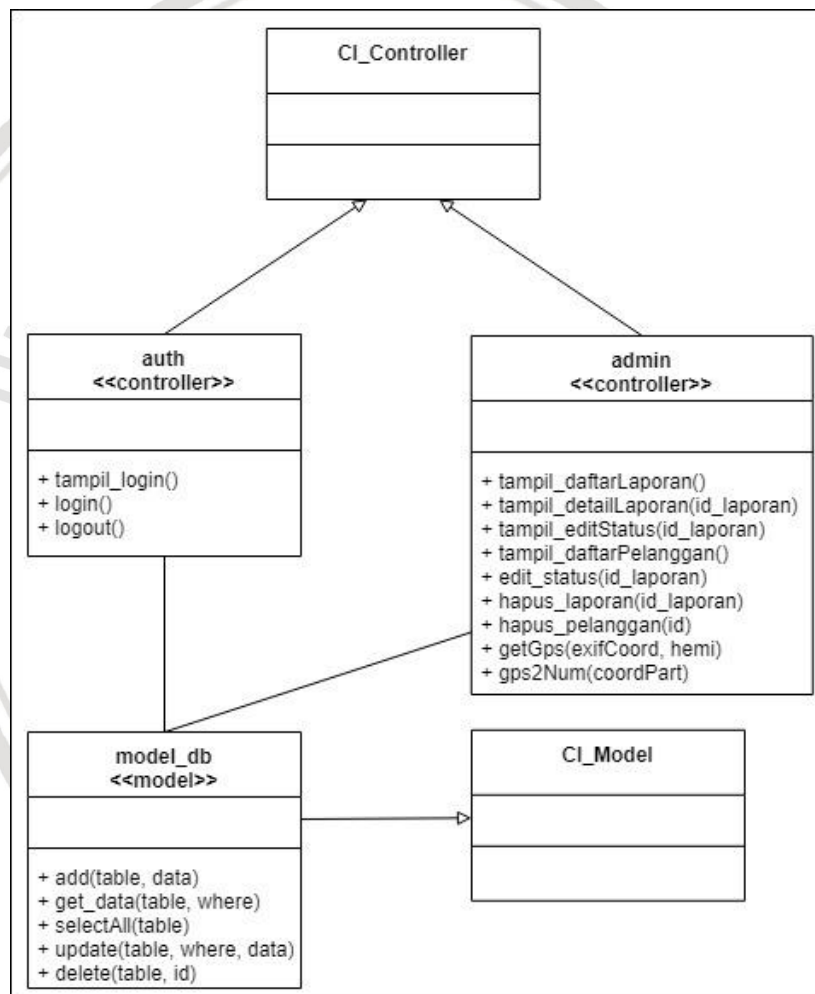
Pada saat user menekan menu *gallery* untuk mengambil gambar yang telah tersimpan di dalam *gallery smartphone* maka akan memanggil *method* `onCreateView(inflater, container, savedInstanceState)` pada *boundary* `GalleryFragment` dan mengeksekusi `PhotoGalleryAsyncLoader(getActivity())` untuk menampilkan gambar yang ada pada *gallery*. Ketika user memilih gambar maka akan menjalankan *method* `onItemClick(parent, view, position, id)` pada *boundary* `GalleryFragment` yang akan menghasilkan nilai *location*. Kemudian data foto yang telah dipilih user dikirimkan ke kelas `GalleryAdapter`. Pada kelas ini terdapat 2 alternatif kondisi, ketika *location* bernilai *null* yang berarti tidak terdapat informasi lokasi pada gambar yang dipilih maka akan menampilkan pesan error pada *boundary* `GalleryFragment`. Kemudian pada alternatif kedua yaitu ketika *location* tidak bernilai *null* atau terdapat informasi lokasi pada gambar yang dipilih maka akan membuka halaman form keluhan pada *boundary* `UploadActivity`.





Gambar 5.7 merupakan gambar *class diagram* dari aplikasi pelaporan keluhan pelanggan. Pada *class diagram* ini terdapat 8 kelas *activity* dengan nama *ComplaintDetail*, *EditComplaint*, *LoginActivity*, *MainActivity*, *PostActivity*, *ProfileActivity*, *RegisterActivity*, dan *Upload Activity*, dimana 8 kelas tersebut melakukan *inheritance* dari kelas *AppCompatActivity* dan *Activity*. 3 kelas *controller* dengan nama *AppController*, *ComplaintController*, dan *RegisterController*. 4 kelas *fragment* dengan nama *BaseFragment*, *CameraFragment*, *GalleryFragment*, dan *HomeFragment*, dimana 4 kelas tersebut melakukan *inheritance* dari kelas *BaseFragment* dan *Fragment*, 2 kelas *model* dengan nama *Complaint* dan *User*, 3 kelas *adapter* dengan nama *AdapterGrid*, *AdapterList*, dan *GalleryAdapter*, serta 15 kelas lainnya.

#### 5.1.4.2 Perancangan Class Diagram Website Admin



**Gambar 5.8 Class Diagram Website Admin**

Gambar 5.8 merupakan gambar *class diagram* dari *website admin*. Pada *class diagram* ini terdapat 2 *controller* dengan nama *auth* dan *admin* yang melakukan *inheritance* dari kelas *CI\_Controller*. Kelas *controller* ini berisi operasi yang digunakan untuk mengolah data pada *website admin*. Kemudian terdapat 1 kelas

model dengan nama `model_db` yang melakukan *inheritance* dari kelas `CI_Model`. Kelas model ini digunakan untuk melakukan akses database.

### 5.1.5 Perancangan Algoritme

Pada tahap ini akan dijelaskan algoritme yang akan diimplementasikan pada pengembangan aplikasi *mobile* pelaporan keluhan pelanggan PDAM berbasis android. Terdapat 3 algoritme yang akan dijelaskan, yaitu *method* `onClick()` yang berada pada kelas `UploadActivity`, *method* `onCreateView()` pada kelas `CameraFragment`, dan *method* `getAddress()` pada kelas `GeotagImage`.

#### 5.1.5.1 Perancangan Algoritme *Method* `onClick()`

1	Start
2	category = data bagian
3	description = data deskripsi
4	idUser = nilai idUser
5	if(deskripsi == true)
6	objek upload
7	memanggil method <code>upload(sourceFile, idUser, category, latitude, longitude, description, filePath)</code>
9	else
10	errorMsg()
11	endif
12	end

Pada *pseudocode* *method* `onClick()` yang berada pada kelas `UploadActivity` ini akan dilakukan proses upload keluhan yang telah user isikan pada form pelaporan keluhan. *Method* ini dimulai dengan inisialisasi variable yang berisi data keluhan yaitu bagian dan deskripsi. Kemudian terdapat seleksi kondisi apabila deskripsi tidak kosong atau bernilai true maka akan mengirim data laporan ke *method* `upload()` yang terdapat pada kelas `ComplaintController` dengan parameter berupa data keluhan untuk diolah. Namun apabila deskripsi kosong maka akan menampilkan pesan error.

#### 5.1.5.2 Perancangan Algoritme *Method* `onCreateView()`

1	Start
2	View = layout fragment_native_camera
3	tvLokasi = data lokasi
4	tvKoordinat = data koordinat
5	opened = hasil proses dari method untuk membuka kamera
6	objek <code>mGoogleApiClient</code>
7	if(opened == false)
8	menampilkan pesan eror
9	mengembalikan nilai view
10	endif
11	captureButton = data button capture
12	operasi <code>setOnClickListener</code> untuk mengambil foto dari
13	kamera
14	<code>mLocationRequest</code> = request lokasi update device
15	return view
16	end

Pada *pseudocode* *method* `onCreateView()` yang dipanggil pada kelas `CameraFragment` ini diawali dengan inisialisasi variable yang berisi data lokasi,

koordinat gambar serta hasil dari operasi untuk membuka kamera. Kemudian terdapat seleksi kondisi apabila kamera tidak berhasil terbuka maka akan menampilkan pesan error, sedangkan apabila kamera berhasil terbuka maka akan menjalankan *method* *setOnClickListener()* untuk mengambil foto melalui kamera.

#### 5.1.5.3 Rancangan Algoritme *Method getAddress()*

1	Start
2	Country = data negara
3	Objek Geocoder
4	Objek strAddrBuilder
5	try
6	addressList = data lokasi dari latitude dan longitude yang ada
7	if (addressList != null)
8	data addressList pada list pertama
9	for(i=0 to i<list maksimal)
10	menambah data alamat
11	end for
12	end if
13	end try
14	catch
15	printStackTrace()
16	end catch
17	return data alamat
18	end

Pada *pseudocode* *method getAddress()* pada kelas *geotagImage* ini diawali dengan inisialisasi variabel yang berisi data negara serta alamat dari latitude dan longitude yang ada. Kemudian terdapat seleksi kondisi apabila data alamat tidak bernilai null maka data alamat akan ditambahkan ke dalam list sampai, namun apabila bernilai null maka akan menampilkan error.

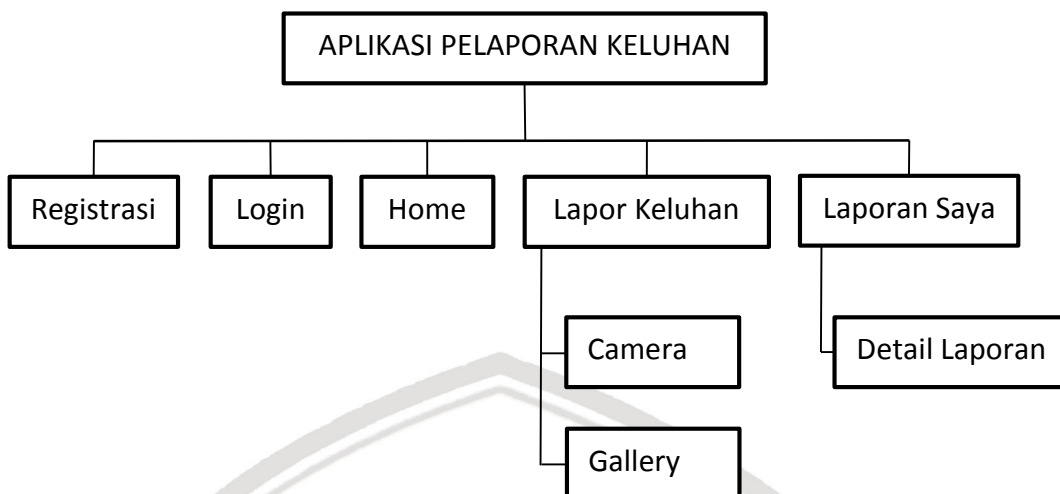
#### 5.1.6 Perancangan Antarmuka

Pada tahap ini akan dijelaskan bagaimana perancangan antarmuka dari Aplikasi Pelaporan Keluhan Pelanggan PDAM sehingga konsistensi dari aplikasi yang akan dibangun tetap terjaga. Pada perancangan antarmuka terbagi menjadi dua jenis yaitu perancangan *screenflow* dan perancangan antarmuka dari masing masing aplikasi mobile yang berbasis android dan website untuk admin.

##### 5.1.6.1 Perancangan *Screenflow*

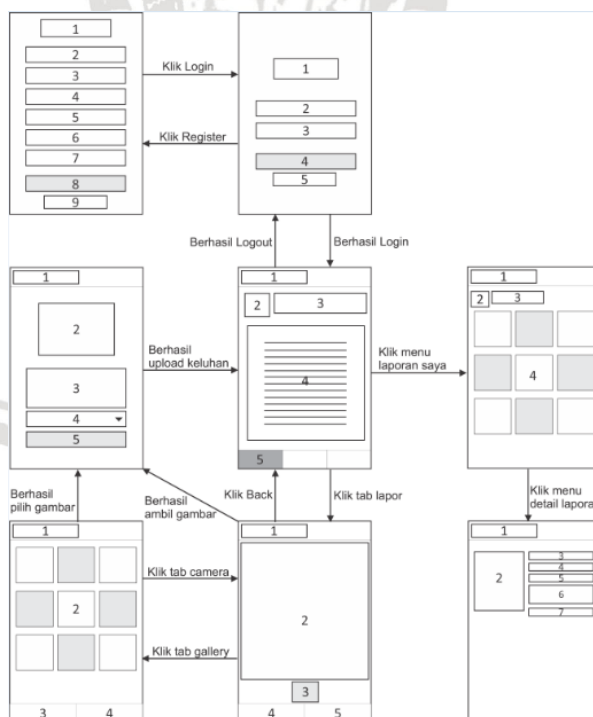
Pada perancangan *screenflow* akan digambarkan bagaimana alur perpindahan dari setiap halaman yang ada pada Aplikasi Pelaporan Keluhan Pelanggan PDAM. Sebelum membuat *screenflow*, akan dibuat *sitemap* aplikasi yang bertujuan untuk memberikan gambaran struktur halaman yang ada pada Aplikasi Pelaporan Keluhan Pelanggan PDAM.

### A. Screenflow Aplikasi Mobile User



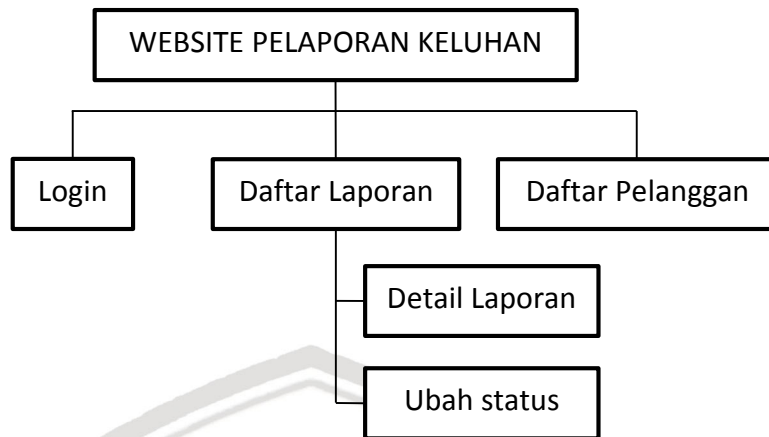
**Gambar 5.9 Sitemap Aplikasi**

Berdasarkan sitemap pada Gambar 5.9 terdapat 5 Halaman Utama pada aplikasi pelaporan keluhan pelanggan PDAM yaitu Registrasi, *Login*, *Home*, Lapor Keluhan, dan Laporan Saya. Kemudian terdapat 3 halaman lain yang dapat diakses ketika user telah membuka halaman utama, yaitu halaman *camera*, *gallery* serta halaman detail laporan. *Sitemap* tersebut akan digunakan sebagai acuan dalam membuat perancangan *screenflow* serta perancangan antarmuka pada aplikasi pelaporan keluhan pelanggan PDAM. Pada Gambar 5.10 akan ditunjukkan alur halaman (*screenflow*) pada aplikasi pelaporan keluhan pelanggan PDAM.



**Gambar 5.10 Screenflow Aplikasi**

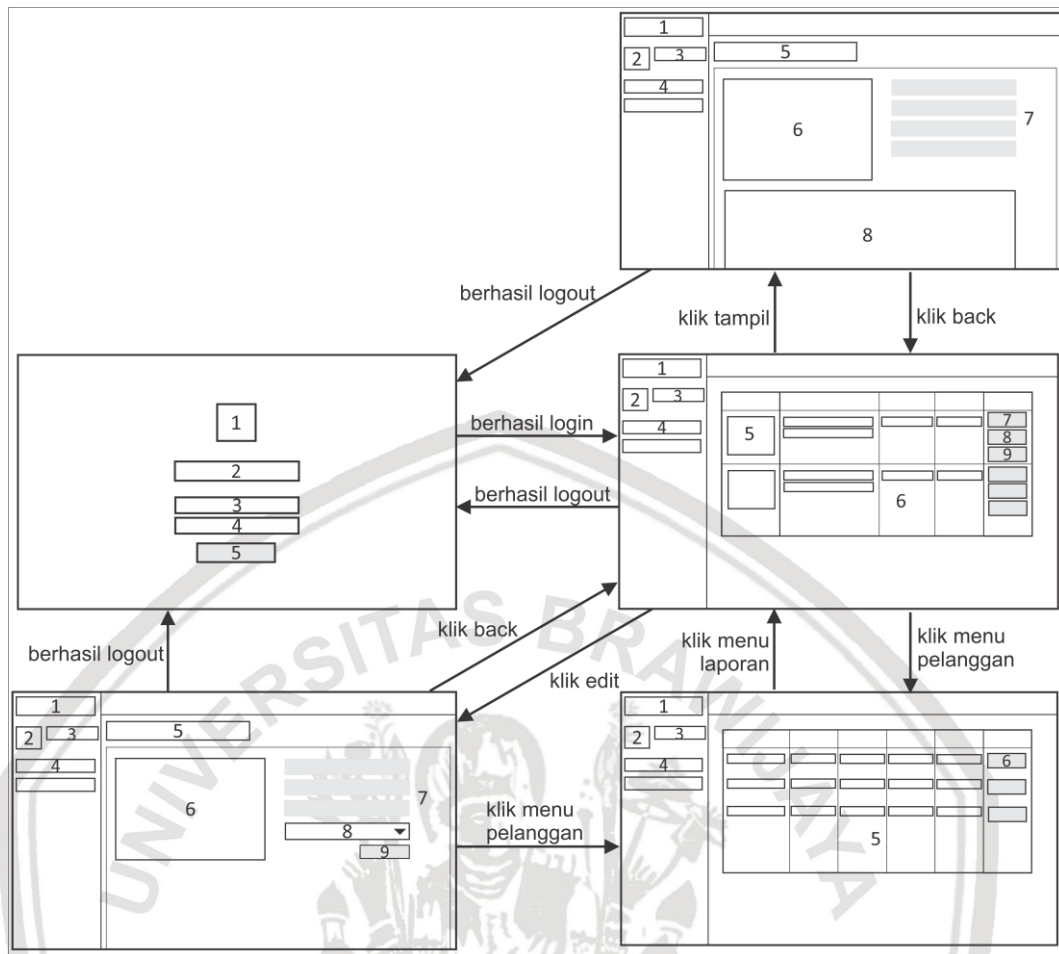
## B. *Screenflow Website Admin*



**Gambar 5.11 Sitemap Website**

Berdasarkan sitemap pada Gambar 5.11 terdapat 3 Halaman Utama pada website pelaporan keluhan pelanggan PDAM yaitu Login, Daftar Laporan, dan Daftar Pelanggan. Kemudian terdapat 2 halaman lain yang dapat diakses ketika admin telah membuka halaman utama, yaitu halaman detail laporan serta halaman ubah status. Sitemap tersebut akan digunakan sebagai acuan dalam membuat perancangan *screenflow* serta perancangan antarmuka pada website pelaporan keluhan pelanggan PDAM. Pada Gambar 5.12 akan ditunjukkan alur halaman (*screenflow*) pada website pelaporan keluhan pelanggan PDAM.





**Gambar 5.12 Screenflow Website Admin**

#### 5.1.6.2 Perancangan Antarmuka

Perancangan antarmuka dibuat untuk menggambarkan detail dari setiap halaman yang terdapat pada Aplikasi Pelaporan Keluhan Pelanggan PDAM yang telah didefinisikan pada *screenflow*. Terdapat dua jenis perancangan antarmuka yaitu perancangan antarmuka untuk aplikasi yang berbasis *mobile* serta antarmuka untuk *website* admin.

## A. Perancangan Antarmuka Aplikasi *Mobile User*

### 1. Halaman *Login*

The diagram shows a vertical stack of five rectangular boxes. Box 1 is at the top, followed by box 2, then box 3, then box 4, and finally box 5 at the bottom. Box 4 is highlighted with a grey background, while the others are white with black borders.

Keterangan:

1. Label "Login"
2. *Textfield Username*
3. *Textfield Password*
4. *Button "Login"*
5. Label "Belum punya akun? Register Sekarang"

**Gambar 5.13 Antarmuka Halaman Login**

Halaman *Login* digunakan oleh pelanggan PDAM sebagai user untuk masuk kedalam akun. Berdasarkan Gambar 5.13 terdapat 2 Label dengan tulisan "Login" sebagai petunjuk bahwa halaman ini merupakan halaman Login dan tulisan "Belum punya akun? Register Sekarang" untuk user yang belum memiliki akun. Selain itu terdapat 2 *textfield* yang dapat digunakan user untuk memasukkan *username* dan *password* serta 1 *Button* dengan tulisan "Login" untuk menjalankan fungsi login dan submit data login kedalam server.

### 2. Halaman Registrasi

The diagram shows a vertical stack of nine rectangular boxes. Box 1 is at the top, followed by boxes 2 through 7, then box 8, and finally box 9 at the bottom. Box 8 is highlighted with a grey background, while the others are white with black borders.

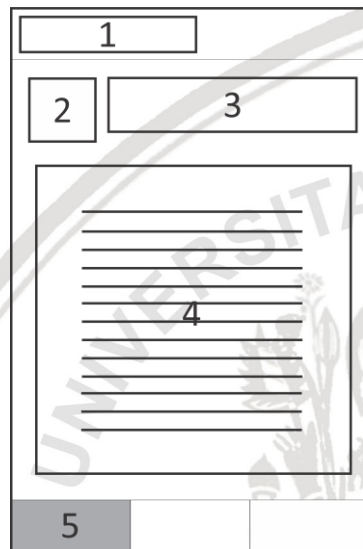
Keterangan:

1. Label "Registrasi"
2. *Textfield "Nama Lengkap"*
3. *Textfield "Nomor Pelanggan"*
4. *Textfield "username"*
5. *Textfield "e-mail"*
6. *Textfield "password"*
7. *Textfield "no HP"*
8. *Button "Register"*
9. Label "Sudah mendaftar? Silahkan Login"

**Gambar 5.14 Antarmuka Halaman Registrasi**

Halaman Registrasi digunakan oleh pelanggan PDAM sebagai user untuk mendaftar guna mendapatkan akun Aplikasi Pelaporan Keluhan Pelanggan PDAM. Berdasarkan Gambar 5.14 terdapat 2 Label dengan tulisan “Registrasi” sebagai petunjuk bahwa halaman ini merupakan halaman Registrasi dan tulisan “Sudah mendaftar? Silahkan Login” untuk user yang sudah memiliki akun. Selain itu terdapat 6 *textfield* yang digunakan user untuk mengisi data diri registrasi yaitu berupa nama lengkap, nomor pelanggan, *username*, e-mail, *password*, serta no Hp. Kemudian terdapat 1 *Button* dengan tulisan “Register” untuk menjalankan fungsi registrasi dan submit data registrasi kedalam server.

### 3. Halaman Home



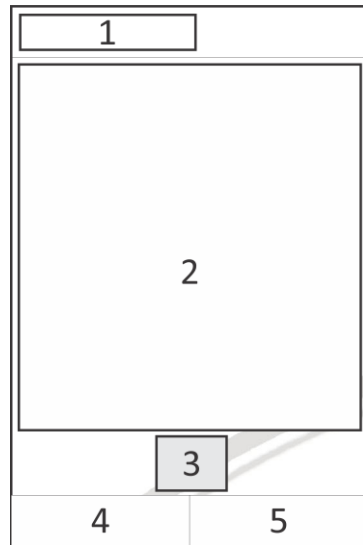
Keterangan:

1. Label “Aplikasi Pelaporan”
2. Gambar Logo PDAM Tirta Tuah Benua Kutai Timur
3. Label “Informasi Pelaporan Keluhan Pelanggan”
4. Label yang berisi informasi tentang pelaporan keluhan
5. Button tab menu

**Gambar 5.15 Antarmuka Halaman Home**

Halaman Home merupakan halaman utama dari aplikasi ini ketika user telah berhasil login. Halaman home digunakan oleh user untuk melihat informasi yang berkaitan dengan pelaporan keluhan pelanggan PDAM. Berdasarkan Gambar 5.15 terdapat 3 label yang berisi tulisan “Aplikasi Pelaporan” sebagai nama aplikasi, tulisan “Informasi Pelaporan Keluhan Pelanggan” sebagai petunjuk bahwa halaman Home memuat suatu informasi mengenai pelaporan keluhan, serta label dengan tulisan informasi. Kemudian terdapat gambar yang berupa gambar dari logo PDAM Tirta Tuah Benua Kutai Timur. Setelah itu terdapat 2 Button yang merupakan tab-tab menu yang ada pada aplikasi pelaporan keluhan pelanggan PDAM. Terdapat 2 menu yaitu Home, Lapor.

#### 4. Halaman *Camera*



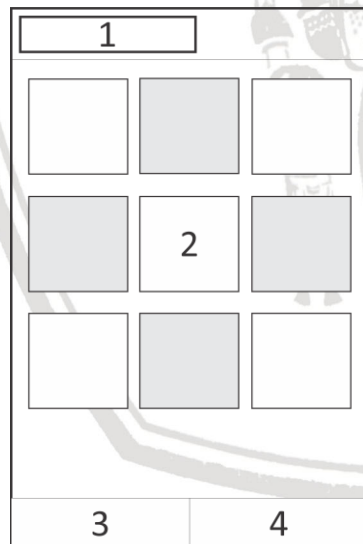
Keterangan:

1. Label “Aplikasi Pelaporan”
2. *Camera*
3. *Button Capture*
4. *Button Tab Camera*
5. *Button Tab Gallery*

**Gambar 5.16 Antarmuka Halaman *Camera***

Halaman *Camera* digunakan pelanggan sebagai user untuk mengambil gambar yang terkait dengan laporan. Berdasarkan Gambar 5.16 terdapat label dengan tulisan “Aplikasi Pelaporan” sebagai nama aplikasi. Terdapat *button capture* untuk mengambil gambar. Kemudian terdapat 2 *button tab* untuk berganti ke menu *camera* dan *gallery*.

#### 5. Halaman *Gallery*



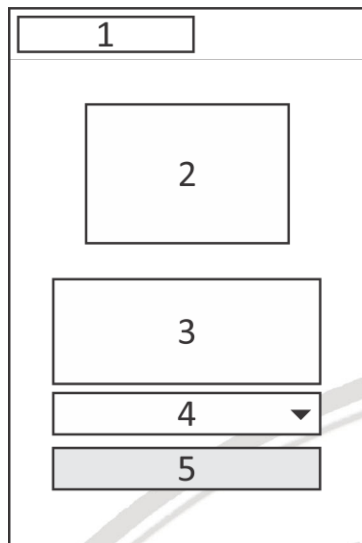
Keterangan:

1. Label “Aplikasi Pelaporan”
2. Gambar yang ada pada *gallery*
3. *Button Tab Camera*
4. *Button Tab Gallery*

**Gambar 5.17 Antarmuka Halaman *Gallery***

Halaman *Gallery* digunakan pelanggan sebagai user untuk memilih gambar yang terkait dengan laporan yang terdapat pada galeri *smartphone* pengguna. Berdasarkan Gambar 5.17 terdapat label dengan tulisan “Aplikasi Pelaporan” sebagai nama aplikasi. Kemudian terdapat gambar yang ada pada galeri pengguna yang ditampilkan dalam bentuk *thumbnail*. Lalu terdapat 2 *button tab* untuk berganti ke menu *camera* dan *gallery*.

## 6. Halaman Lapor Keluhan



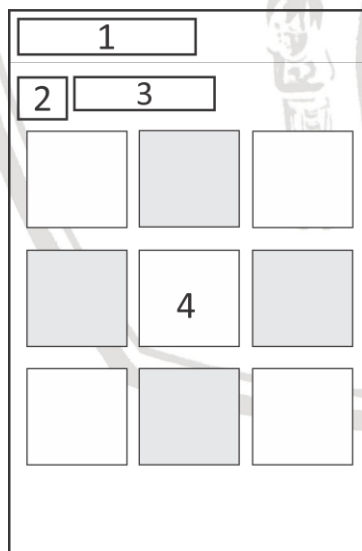
Keterangan:

1. Label “Aplikasi Pelaporan”
2. Gambar Laporan
3. *Textfield* “Deskripsi Laporan”
4. *Dropdown* “Bagian Keluhan”
5. *Button* “Submit”

**Gambar 5.18 Antarmuka Halaman Lapor**

Halaman Lapor digunakan oleh pelanggan sebagai user untuk dapat mengisi deskripsi gambar yang telah diambil untuk dilaporkan. Berdasarkan Gambar 5.18 terdapat 1 label dengan tulisan “Aplikasi Pelaporan” sebagai nama aplikasi. Kemudian terdapat gambar laporan yang telah diambil sebelumnya. Selain itu terdapat 1 *textfield* serta 1 *dropdown* yang digunakan untuk memasukkan keterangan foto deskripsi serta bagian laporan. Terdapat 1 *button* yaitu button “submit” untuk melakukan submit data laporan ke dalam server.

## 7. Halaman Laporan Saya



Keterangan:

1. Label “Aplikasi Pelaporan”
2. Gambar *profile picture*
3. Label *username* pelanggan
4. Gambar laporan yang telah diupload

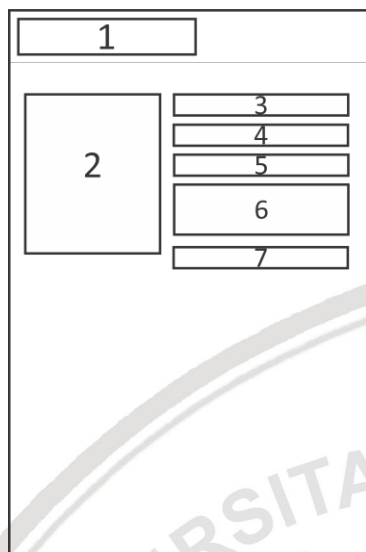
**Gambar 5.19 Antarmuka Halaman Laporan Saya**

Halaman Laporan Saya digunakan oleh pelanggan sebagai user untuk melihat daftar laporan yang telah dilaporkan sebelumnya. Berdasarkan Gambar 5.19 terdapat 2 Label yaitu label dengan tulisan “Aplikasi Pelaporan” sebagai nama aplikasi dan label *username* pelanggan. Kemudian terdapat 2 gambar yaitu gambar



*profile picture* dan gambar laporan yang telah dilaporkan sebelumnya yang ditampilkan dalam bentuk *thumbnail*.

## 8. Halaman Detail Laporan



Keterangan:

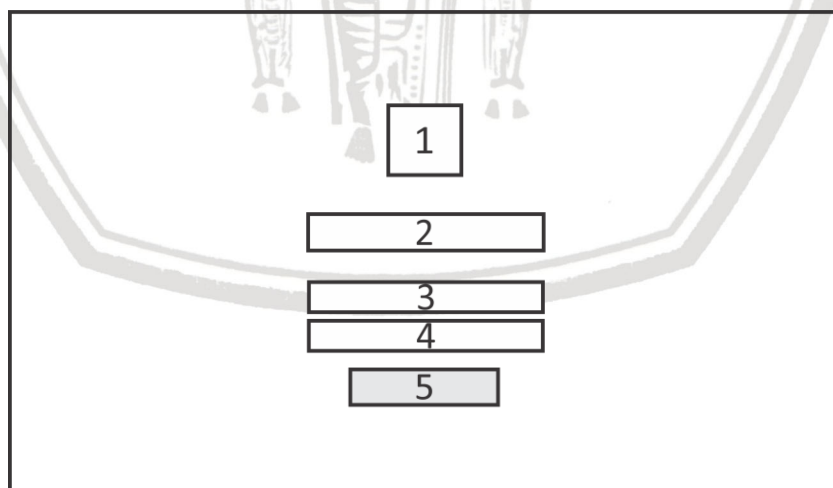
1. Label "Aplikasi Pelaporan"
2. Gambar laporan
3. Label username
4. Label tanggal lapor
5. Label bagian laporan
6. Label deskripsi laporan
7. Label status laporan

**Gambar 5.20 Antarmuka Halaman Detail Laporan**

Halaman detail laporan digunakan oleh pelanggan sebagai user untuk melihat detail dari laporan yang telah dilaporkan sebelumnya. Berdasarkan Gambar 5.20 terdapat 6 label yaitu label yang berisi tulisan "Aplikasi Pelaporan" sebagai nama aplikasi dan keterangan laporan yang berupa username, tanggal lapor, bagian, deskripsi serta status laporan. Kemudian terdapat gambar dari laporan yang telah diupload.

## B. Perancangan Antarmuka Website Admin

### 1. Halaman Login Admin



**Gambar 5.21 Antarmuka Halaman Login Admin**

Keterangan:

1. Gambar Logo PDAM Tirta Tuah Benua Kutai Timur

2. Label "Login"
3. *Textfield* "username"
4. *Textfield* "password"
5. *Button* "login"

Halaman Login digunakan oleh karyawan PDAM sebagai admin untuk masuk kedalam akun. Berdasarkan Gambar 5.21 terdapat gambar berupa logo dari PDAM Tirta Tuah Benua, kemudian terdapat 1 Label dengan tulisan "Login". Selain itu terdapat 2 *textfield* yang dapat digunakan admin untuk memasukkan *username* dan *password* serta 1 *Button* dengan tulisan "Login" untuk menjalankan fungsi login dan submit data login kedalam server.

## 2. Halaman Daftar Laporan Admin

**Gambar 5.22 Antarmuka Halaman Daftar Laporan Admin**

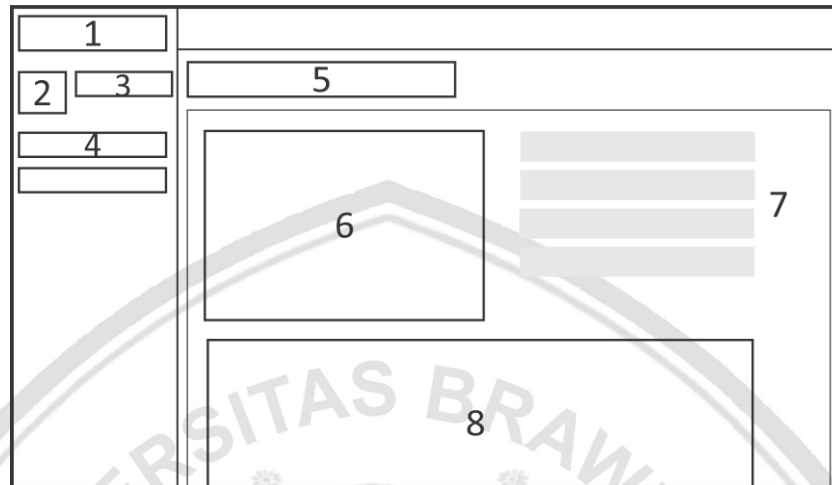
Keterangan:

1. Label "Pelaporan Keluhan Pelanggan PDAM"
2. Gambar Logo PDAM Tirta Tuah Benua Kutai Timur
3. Label nama admin
4. *Button* tab menu
5. Gambar laporan keluhan pelanggan
6. Label deskripsi, bagian dan status dari laporan
7. *Button* "tampil"
8. *Button* "ubah"
9. *Button* "hapus"

Halaman Daftar Laporan Admin digunakan oleh karyawan sebagai admin untuk melihat daftar laporan yang telah dilaporkan oleh user. Berdasarkan Gambar 5.22 terdapat 3 Label dengan tulisan "Pelaporan Keluhan Pelanggan PDAM", nama dari admin serta deskripsi, bagian dan status laporan. lalu terdapat 4 jenis *button*, yaitu

2 *button* tab menu yang ada di website pelaporan keluhan, *button* tampil untuk menampilkan detail laporan, *button* hapus untuk menghapus laporan yang dinilai tidak sesuai, dan *button* ubah untuk merubah status dari laporan tersebut. Kemudian terdapat Gambar laporan yang telah dilaporkan oleh user.

### 3. Halaman Detail Laporan Admin



**Gambar 5.23 Antarmuka Halaman Detail Laporan Admin**

Keterangan:

1. Label “Pelaporan Keluhan Pelanggan PDAM”
2. Gambar Logo PDAM Tirta Tuah Benua Kutai Timur
3. Label nama admin
4. *Button* tab menu
5. Label “Detail Laporan”
6. Gambar Laporan Keluhan
7. Label keterangan gambar
8. Google Maps

Halaman Detail Laporan Admin digunakan oleh karyawan sebagai admin untuk melihat secara detail laporan yang telah dilaporkan oleh user. Berdasarkan Gambar 5.23 terdapat 4 Label dengan tulisan “Pelaporan Keluhan Pelanggan PDAM” dan “Detail Laporan”, label nama admin, serta label keterangan gambar yang berupa nama pelapor, deskripsi, bagian yang menangani laporan dan status laporan saat ini. Kemudian terdapat 2 *button* tab menu yang ada di website pelaporan keluhan. Setelah itu terdapat 2 gambar yaitu logo PDAM Tirta Tuah Benua dan gambar laporan keluhan serta Google Maps untuk melihat lokasi laporan tersebut

#### 4. Halaman Ubah Status

**Gambar 5.24 Antarmuka Halaman Ubah Status**

Keterangan:

1. Label “Pelaporan Keluhan Pelanggan PDAM”
2. Gambar Logo PDAM Tirta Tuah Benua Kutai Timur
3. Label nama admin
4. *Button* tab menu
5. Label “Edit Status Laporan”
6. Gambar Laporan Keluhan
7. Label keterangan gambar
8. *Dropdown* “Status”
9. *Button* “Submit”

Halaman Ubah Status digunakan oleh karyawan sebagai admin untuk merubah status dari laporan yang telah dilaporkan oleh user Berdasarkan Gambar 5.24 terdapat 4 Label dengan tulisan “Pelaporan Keluhan Pelanggan PDAM” dan “Edit Status Laporan”, label nama admin, serta label keterangan gambar yang berupa nama pelapor, deskripsi, dan bagian yang menangani laporan. Kemudian terdapat 2 *button* tab menu yang ada di website pelaporan keluhan. Setelah itu terdapat 2 gambar yaitu logo PDAM Tirta Tuah Benua dan gambar laporan keluhan serta Google Maps untuk melihat lokasi laporan tersebut. Untuk dapat merubah status dari laporan, disediakan 1 *dropdown* yang berisi pilihan dari status laporan saat ini yang dapat dipilih oleh admin serta *button submit* untuk menyimpan hasil perubahan status.

## 5. Halaman Daftar Pelanggan

**Gambar 5.25 Antarmuka Halaman Daftar Pelanggan**

Keterangan:

1. Label “Pelaporan Keluhan Pelanggan PDAM”
2. Gambar Logo PDAM Tirta Tuah Benua Kutai Timur
3. Label nama admin
4. *Button* tab menu
5. Label data diri pelanggan
6. *Button* “hapus”

Halaman Daftar Pelanggan Admin digunakan oleh karyawan sebagai admin untuk melihat daftar pelanggan yang telah terdaftar pada aplikasi. Berdasarkan Gambar 5.25 terdapat 3 Label dengan tulisan “Pelaporan Keluhan Pelanggan PDAM”, nama dari admin serta data diri pelanggan. Lalu terdapat 2 jenis *button*, yaitu 2 *button* tab menu yang ada di website pelaporan keluhan, *button* hapus untuk menghapus pelanggan.

## 5.2 Implementasi

### 5.2.1 Spesifikasi Sistem

Pada tahap implementasi dibutuhkan perangkat baik berupa perangkat lunak maupun perangkat keras sebagai media implementasi. Pada spesifikasi sistem akan dijelaskan mengenai spesifikasi perangkat lunak serta spesifikasi perangkat keras yang digunakan dalam pembuatan Aplikasi Pelaporan Keluhan Pelanggan PDAM.

#### 5.2.1.1 Spesifikasi Perangkat Keras

Proses pengembangan Aplikasi Pelaporan Keluhan Pelanggan PDAM dilakukan dengan menggunakan laptop dan *smartphone*. Laptop akan digunakan untuk merepresentasikan perancangan menjadi kode program. *Smartphone* akan



digunakan untuk menjalankan kode program dari Aplikasi pelaporan Keluhan Pelanggan PDAM. Pada Tabel 5.4 dan Tabel 5.5 didefinisikan Spesifikasi dari perangkat keras berupa laptop dan *smartphone* yang digunakan dalam tahap implementasi.

**Tabel 5.4 Spesifikasi Perangkat Keras Laptop**

Keterangan	Spesifikasi
<i>System Model</i>	Toshiba Satellite L645
<i>Processor</i>	Intel(R) Core(TM) i5 CPU M 480 @ 2.67GHz
<i>Memory</i>	2048MB RAM

**Tabel 5.5 Spesifikasi Perangkat Keras *Smartphone***

Keterangan	Spesifikasi
<i>System Model</i>	Oppo F1S
<i>Processor</i>	Octa Core
<i>Memory</i>	3.0GB RAM

#### 5.2.2.2 Spesifikasi Perangkat Lunak

Pada proses pengembangan Aplikasi Pelaporan Keluhan Pelanggan PDAM ini digunakan beberapa perangkat lunak untuk mengimplemantasikan perancangan yang telah dibuat kedalam bentuk kode program. Pada Tabel 5.6 didefinisikan perangkat lunak yang digunakan dalam pengembangan Aplikasi Pelaporan Keluhan Pelanggan PDAM.

**Tabel 5.6 Spesifikasi Perangkat Lunak**

Keterangan	Spesifikasi
<i>Operating System</i>	Windows 8.1 64-bit
<i>Programing Language</i>	Java, PHP
<i>Text Editor</i>	Android Studio versi 2.3.3, Sublime
<i>Server</i>	Apache

#### 5.2.2 Batasan Implementasi

Terdapat beberapa batasan pada Implementasi Aplikasi Pelaporan Keluhan Pelanggan PDAM, antara lain:

1. Sistem dijalankan pada perangkat yang harus tersambung dengan koneksi internet
2. Sistem dijalankan pada perangkat bergerak dengan sistem operasi Android

3. Pertukaran data yang dilakukan antara aplikasi dan server menggunakan format JSON.

### 5.2.3 Implementasi Sistem

Pada tahap implemenasi ini akan dilakukan berdasarkan pada perancangan sistem yang sebelumnya dibuat. Tahapan pertama yang dilakukan pada saat implementasi yaitu mengimplementasikan rancangan basis data menggunakan MySQL.

#### 5.2.3.1 Implementasi Basis Data

##### A. Data Definition Language

Tabel 5.7 sampai Tabel 5.9 merupakan implementasi dari perancangan tabel-tabel pada basis data yang telah di definisikan sebelumnya. Tabel 5.7 merupakan *syntax* pembuatan tabel member dimana terdapat 1 kolom dengan tipe data integer yaitu `id_member`, 4 kolom dengan tipe data varchar yaitu `nama`, `identitas`, `email`, dan `no_hp`, serta 1 kolom dengan tipe data datetime yaitu `tgl_registrasi`.

**Tabel 5.7 Syntax Create Tabel Member**

```
CREATE TABLE `member` (
  `id_member` int(11) NOT NULL PRIMARY KEY,
  `nama` varchar(255) NOT NULL,
  `no_identitas` varchar(25) NOT NULL,
  `email` varchar(255) NOT NULL,
  `no_hp` varchar(12) NOT NULL,
  `tgl_registrasi` datetime NOT NULL
)
```

Tabel 5.8 merupakan *syntax* pembuatan tabel user dimana terdapat 1 kolom dengan tipe data integer yaitu `id_user` dan 3 kolom dengan tipe data varchar yaitu `username`, `password`, dan `level`.

**Tabel 5.8 Syntax Create Tabel User**

```
CREATE TABLE `user` (
  `id_user` int(11) NOT NULL PRIMARY KEY,
  `username` varchar(20) NOT NULL,
  `password` varchar(32) NOT NULL,
  `level` varchar(6) NOT NULL
)
```

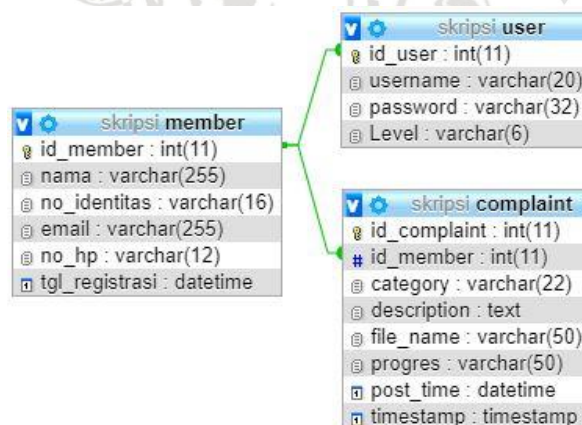
Tabel 5.9 Merupakan *syntax* pembuatan tabel laporan dimana terdapat 2 kolom dengan tipe data integer yaitu `id_laporan` dan `id_member`, 3 kolom dengan tipe data varchar yaitu `bagian`, `file_name`, `progress`, 1 kolom dengan tipe text yaitu

deskripsi, 1 kolom dengan tipe data datetime yaitu post\_time, dan 1 kolom dengan tipe data timestamp yaitu time\_stamp.

**Tabel 5.9 Syntax Create Tabel Complaint**

```
CREATE TABLE `complaint` (
  `id_complaint` int(11) PRIMARY KEY NOT NULL,
  `id_member` int(11) NOT NULL,
  `category` varchar(22) NOT NULL,
  `description` text NOT NULL,
  `file_name` varchar(50) NOT NULL,
  `progres` varchar(22) NOT NULL,
  `post_time` datetime NOT NULL,
  `timestamp` timestamp NOT NULL DEFAULT '0000-00-00
00:00:00' ON UPDATE CURRENT_TIMESTAMP,
  CONSTRAINT FK_ID_MEMBER FOREIGN KEY (id_member)
REFERENCES member(id_member)
)
```

## B. Physical Data Model



**Gambar 5.26 Physical Data Model basis data**

Gambar 5.26 adalah gambaran *Physical Data Model* basis data dari Aplikasi Pelaporan Keluhan Pelanggan PDAM. *Physical Data Model* adalah hasil normalisasi dari prancangan ERD yang sebelumnya telah didefinisikan pada sub bab 5.1.2. pada basis data yang telah dibuat terdapat 3 tabel degan nama member, user, dan complaint. Terdapat dua relasi dan kardinalitas antar tabel yaitu tabel member dan complaint serta tabel member dan user. Pada tabel member dan complaint terdapat relasi yang dihubungkan dengan id\_member pada tabel member sebagai *primary key* serta id\_member pada tabel complaint sebagai *foreign key*. Kemudian pada tabel member dan user terdapat relasi yang dihubungkan dengan id\_member yang terletak di tabel member sebagai *primary key* serta id\_user pada tabel user sebagai *foreign key*.

5.2.3.2 Implementasi Kode Program

A. Implementasi Kode Program Fungsi Melaporkan Keluhan pada Kelas

UploadActivity

Nama *Method* : onClick()

Source Code :

```

1 public void onClick(View view) {
2     /* ALGORITME UNGGAH FOTO */
3     category = tempCategory;
4     description = etDeskripsi.getText().toString().trim();
5     idUser = session.getIdUser();
6     if (deskripsi && judulbool) {
7         ComplaintController upload = new
8         ComplaintController(UploadActivity.this);
9         upload.upload(sourceFile,idUser,category,latitude,lo
10         ngitude,description, filePath);
11     }
12     else {
13         errorMsg();
14     }
15 }
```

B. Implementasi Kode Program Fungsi Mengambil Gambar Melalui Kamera pada Kelas CameraFragment

Nama *method* : onCreateView()

Source Code :

```

1 public View onCreateView(LayoutInflater inflater, ViewGroup
2 container,
3         Bundle savedInstanceState) {
4     View view =
5     inflater.inflate(R.layout.fragment_native_camera,
6     container, false);
7     tvLokasi =
8     (TextView)view.findViewById(R.id.tvLokasi);
9     tvKoordinat =
10    (TextView)view.findViewById(R.id.tvKoordinat);
11    boolean opened = safeCameraOpenInView(view);
12    mGoogleApiClient = new
13    GoogleApiClient.Builder(mContext)
14        .addConnectionCallbacks(this)
15        .addOnConnectionFailedListener(this)
16        .addApi(LocationServices.API)
17        .build();
18    if(opened == false){
19        Log.d("CameraGuide", "Error, Camera failed to
20 open");
21        return view;
22    }
23
24 }
```

```

25         ImageButton captureButton = (ImageButton)
26         view.findViewById(R.id.button_capture);
27         captureButton.setOnClickListener(
28             new View.OnClickListener() {
29                 @Override
30                 public void onClick(View v) {
31                     // get an image from the camera
32                     mCamera.takePicture(null, null,
33                     mPicture);
34
35                 }
36             }
37         );
38
39         mLocationRequest = LocationRequest.create()
40
41         .setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY)
42             .setInterval(10 * 1000)
43             .setFastestInterval(1 * 1000);
44
45         return view;
46     }
47
48
49
50

```

### C. Implementasi Kode Program getAddress() pada Kelas GeotagImage

Nama *method* : getAddress()

Source Code :

```

1  public String getAddress(double lat, double longtd, Context
2  context){
3      Locale country = Locale.getDefault();
4      Geocoder geocoder = new
5      Geocoder(context, country);
6      StringBuilder strAddrBuilder = new
7      StringBuilder();
8
9      try{
10         List<Address> addressList =
11         geocoder.getFromLocation(lat, longtd, 1);
12         if(addressList != null){
13             Address fetchedAddress =
14             addressList.get(0);
15             for(int
16             i=0; i<fetchedAddress.getMaxAddressLineIndex(); i++){
17
18             strAddrBuilder.append(fetchedAddress.getAddressLine(i)).app
19             end("\n");
20         }
21     }
22     } catch (IOException e) {
23         e.printStackTrace();
24     }
25     return strAddrBuilder.toString();

```



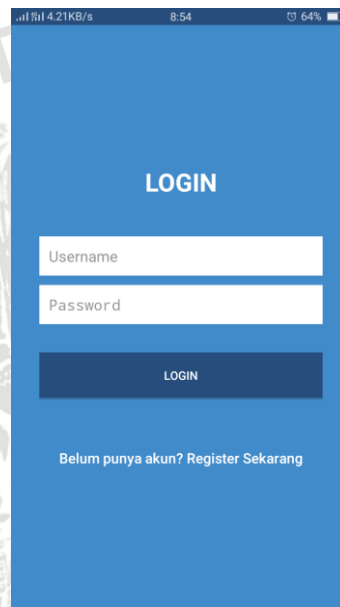
26	}
27	}

### 5.2.3.3 Implementasi Antarmuka

Pada implementasi antarmuka akan diberikan beberapa sampel antarmuka aplikasi pelaporan keluhan pelanggan PDAM seperti halaman login user, halaman home user, halaman kamera, halaman *gallery*, halaman form pelaporan keluhan dll. Implementasi antarmuka aplikasi dibuat berdasarkan perancangan antarmuka pada sub bab 5.1.6 Perancangan Antarmuka.

#### A. Implementasi Antarmuka Halaman *Login User*

Gambar 5.27 menunjukkan implementasi dari antarmuka halaman Login User yang berisi *textfield* untuk memasukkan username dan password serta *button* untuk login.



Gambar 5.27 Implementasi Antarmuka Halaman *Login User*

## B. Implementasi Antarmuka Halaman Registrasi

Gambar 5.28 merupakan implementasi antarmuka halaman registrasi yang berisi 6 *textfield* untuk memasukkan data diri user serta *button* untuk registrasi.

**Gambar 5.28 Implementasi Antarmuka Halaman Registrasi**

## C. Implementasi Antarmuka Halaman *Home User*

Gambar 5.29 menunjukkan implementasi dari antarmuka halaman home user yang menampilkan informasi mengenai aplikasi pelaporan keluhan.



**Gambar 5.29 Implementasi Antarmuka Halaman *Home User***

#### D. Implementasi Antarmuka Halaman *Camera*

Gambar 5.30 merupakan implementasi antarmuka halaman camera yang digunakan untuk mengambil gambar yang akan dilaporkan.



Gambar 5.30 Implementasi Antarmuka Halaman *Camera*

#### E. Implementasi Antarmuka Halaman *Gallery*

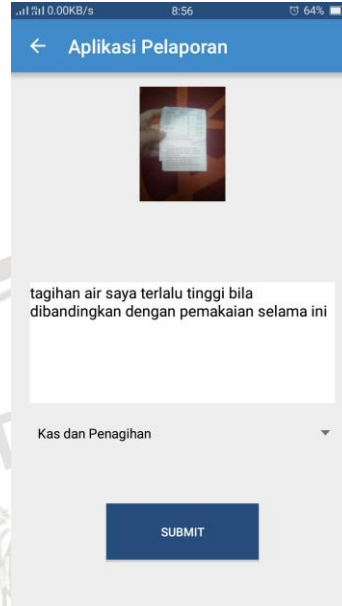
Gambar 5.31 menunjukkan implementasi dari antarmuka halaman *gallery* yang menampilkan gambar pada *device* user untuk dilaporkan.



Gambar 5.31 Implementasi Antarmuka Halaman *Gallery*

#### F. Implementasi Antarmuka Halaman Form Pelaporan Keluhan

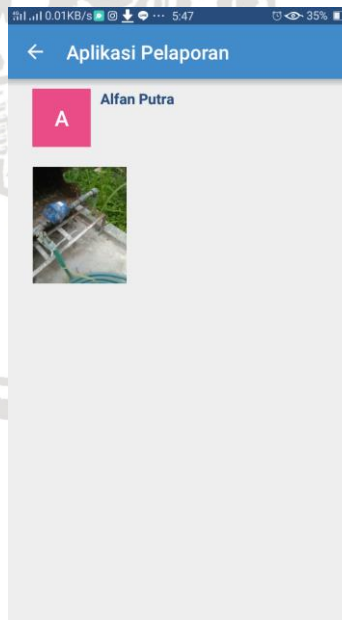
Gambar 5.32 merupakan implementasi antarmuka halaman Form Pelaporan Keluhan yang berisi *textfield* untuk menuliskan deskripsi dari laporan serta memilih bagian penanganan laporan.



Gambar 5.32 Implementasi Antarmuka Halaman Form Pelaporan Keluhan

#### G. Implementasi Antarmuka Halaman Laporan Saya

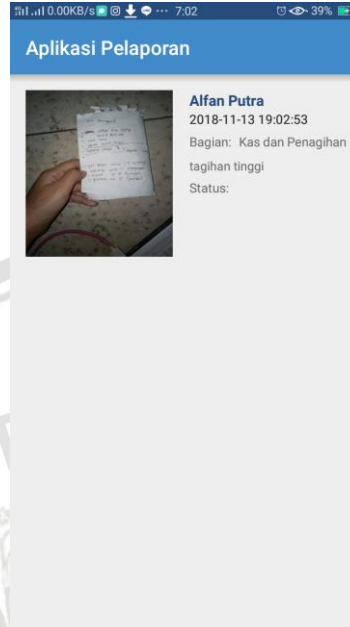
Gambar 5.33 menunjukkan implementasi dari antarmuka halaman laporan saya yang menampilkan gambar dari laporan yang telah dilaporkan.



Gambar 5.33 Implementasi Antarmuka Halaman Laporan Saya

## H. Implementasi Antarmuka Halaman Detail Laporan

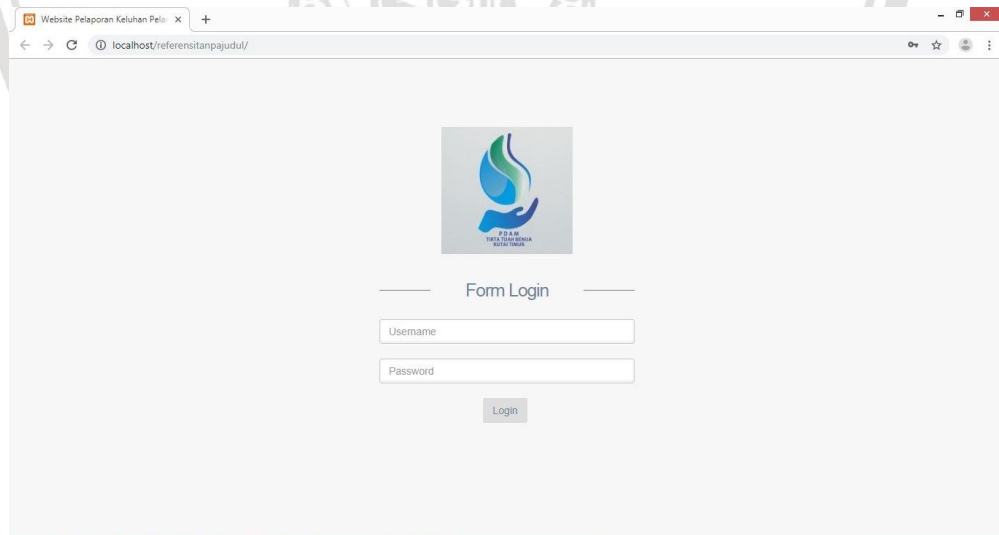
Gambar 5.34 merupakan implementasi dari antarmuka halaman detail laporan yang menampilkan gambar laporan, deskripsi serta status dari laporan yang telah dilaporkan.



Gambar 5.34 Implementasi Antarmuka Halaman Detail Laporan

## I. Implementasi Antarmuka Halaman *Login Admin*

Gambar 5.35 merupakan implementasi dari antarmuka halaman login admin yang berisi *textfield* yang digunakan untuk memasukkan *username* dan *password* serta *button* untuk login admin.

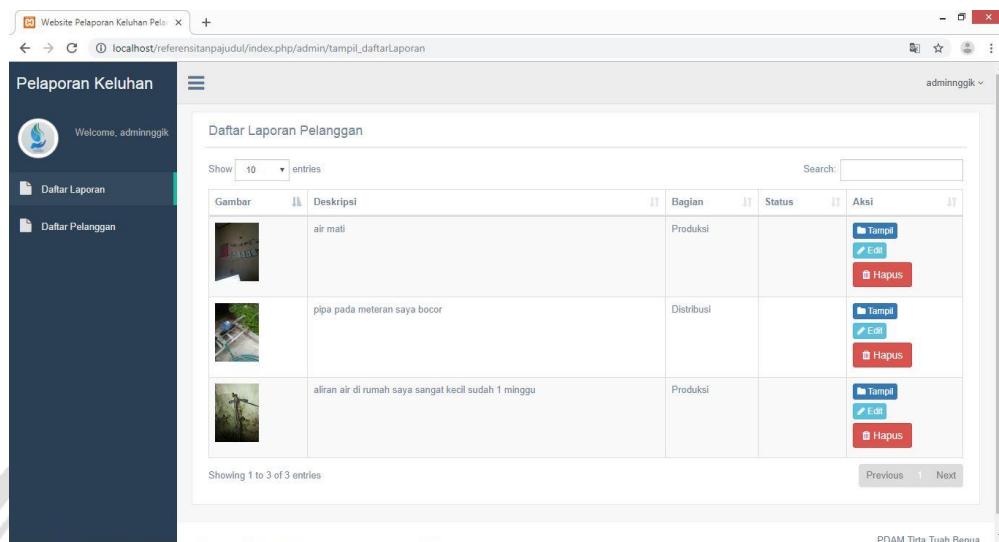


Gambar 5.35 Implementasi Antarmuka Halaman *Login Admin*



## J. Implementasi Antarmuka Halaman Daftar Laporan

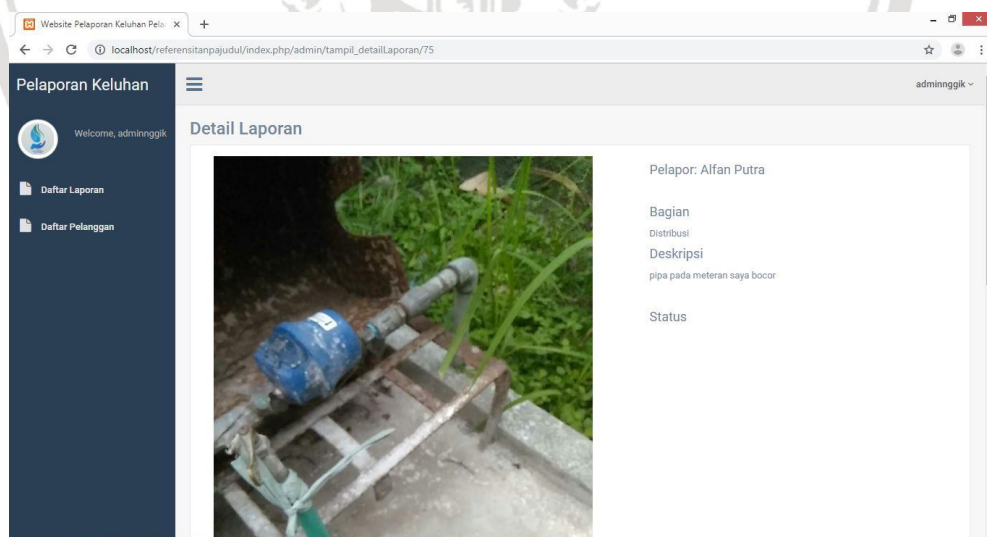
Gambar 5.36 merupakan implementasi dari antarmuka halaman Daftar Laporan yang menampilkan daftar semua laporan yang telah dilaporkan oleh user. Daftar laporan ditampilkan dalam bentuk tabel yang berisi gambar, deskripsi, status, serta aksi yang dapat dilakukan oleh admin.



**Gambar 5.36 Implementasi Antarmuka Halaman Daftar Laporan**

## K. Implementasi Antarmuka Halaman Detail Laporan Admin

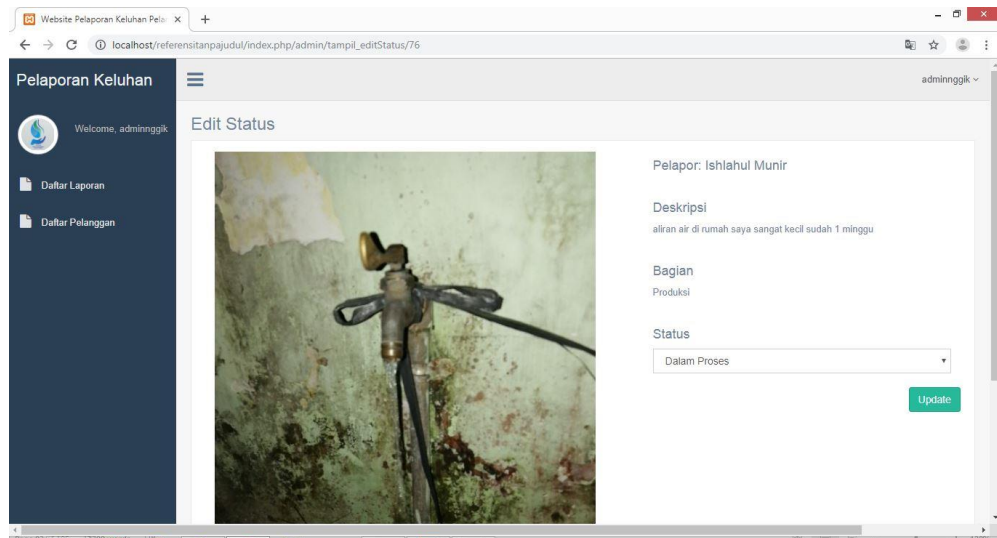
Gambar 5.37 merupakan implementasi antarmuka halaman Detail Laporan Admin yang menampilkan detail dari laporan user. Pada halaman ini ditampilkan gambar, nama pelapor, deskripsi lengkap dari laporan, serta peta yang menampilkan lokasi dari user.



**Gambar 5.37 Implementasi Antarmuka Halaman Detail Laporan Admin**

## L. Implementasi Antarmuka Halaman Ubah Status

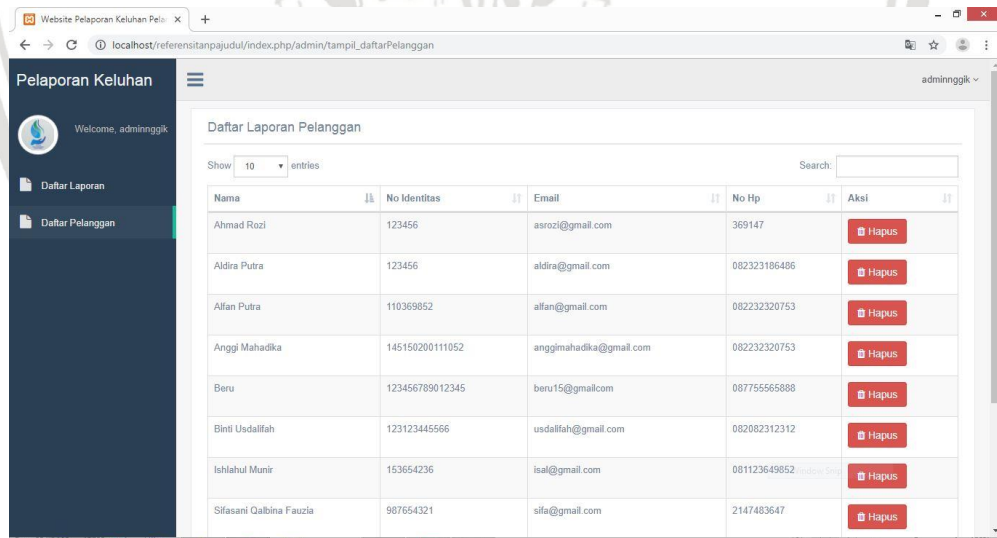
Gambar 5.38 merupakan implementasi antarmuka halaman Ubah Status yang digunakan oleh admin untuk mengganti status dari laporan user.



Gambar 5.38 Implementasi Antarmuka Halaman Ubah Status

## M. Implementasi Antarmuka Halaman Daftar Pelanggan

Gambar 5.39 merupakan implementasi dari antarmuka halaman Daftar Pelanggan yang menampilkan daftar dari pelanggan yang telah terdaftar. Daftar pelanggan ditampilkan dalam bentuk tabel yang berisi nama, no. identitas, email, no hp, dan aksi yang dapat dilakukan oleh admin.



Gambar 5.39 Implementasi Antarmuka Halaman Daftar Pelanggan

## BAB 6 PENGUJIAN DAN ANALISIS

Bab ini berisi pembahasan mengenai pengujian yang dilakukan pada aplikasi pelaporan keluhan pelanggan PDAM berbasis android. Pengujian yang akan dilakukan yaitu pengujian unit, pengujian validasi, serta pengujian *compatibility*.

### 6.1 Pengujian Unit

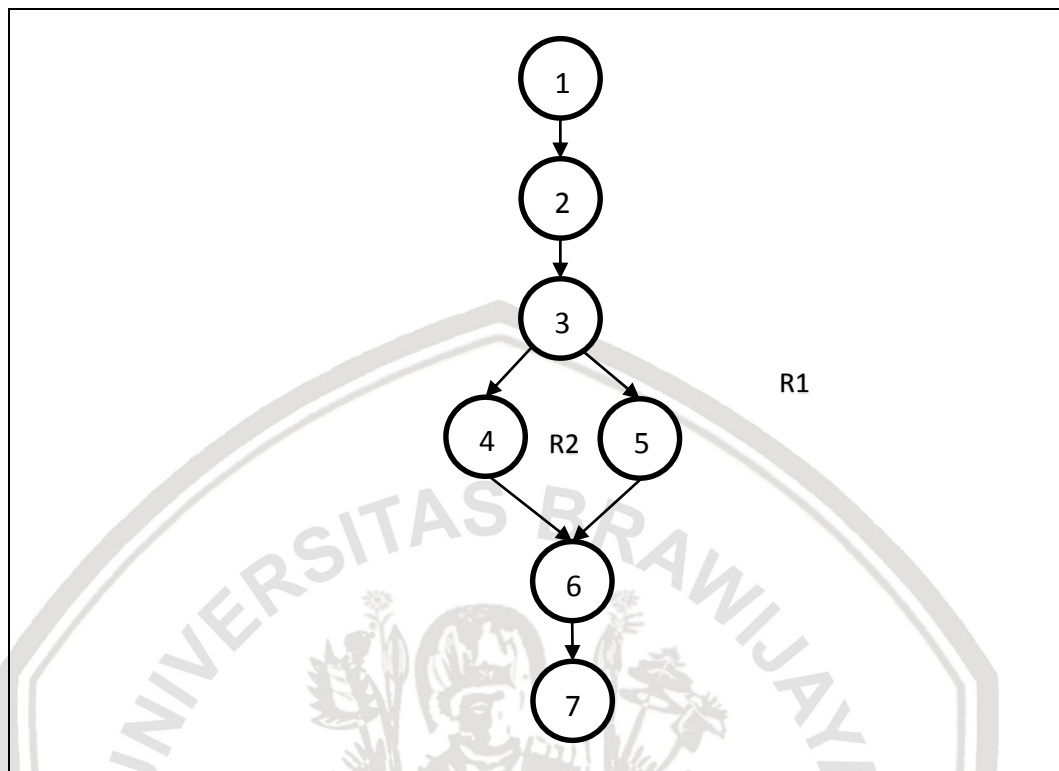
#### 6.1.1 Pengujian Unit operasi onClick() pada kelas UploadActivity.java

##### 1. Pseudocode

```
Start (1)
category = data kategori
description = data deskripsi
idUser = nilai idUser } (2)
if(deskripsi == true) (3)
    objek upload
    memanggil method upload(sourceFile,idUser, category, latitude,
    longitude, description, filePath) } (4)
else
    errorMsg() } (5)
endif (6)
end (7)
```

## 2. Basis Path Testing

### a. Flow Graph



### b. Cyclomatic Complexity

- $V(G) = 2$ , terdapat 2 region yaitu R1 dan R2
- $V(G) = \text{edges} - \text{nodes} + 2$   
 $= 7 - 7 + 2 = 2$
- $V(G) = \text{predicate nodes} + 1$   
 $= 1 + 1 = 2$

### c. Independent Path

- Jalur 1 = 1 – 2 – 3 – 4 – 6 – 7
- Jalur 2 = 1 – 2 – 3 – 5 – 6 – 7

Test case dan hasil pengujian operasi onClick() pada kelas UploadActivity.java akan didefinisikan pada Tabel 6.1

**Tabel 6.1 Hasil Pengujian Unit Operasi onClick() pada Kelas UploadActivity.java**

No	No. Jalur	Prosedur Uji	Expected Result	Result	Hasil
1	1	Valid input, dimana deskripsi bernilai true	Operasi akan mengirimkan data upload ke kelas ComplaintController	Operasi berhasil mengirimkan data upload ke kelas ComplaintController	Valid
2	2	Invalid input, dimana deskripsi bernilai false	Operasi akan menjalankan operasi errorMsg()	Operasi berhasil menjalankan operasi errorMsg()	Valid

### 6.1.2 Pengujian Unit Operasi onCreateView() pada kelas CameraFragment.java

#### 1. Pseudocode

```

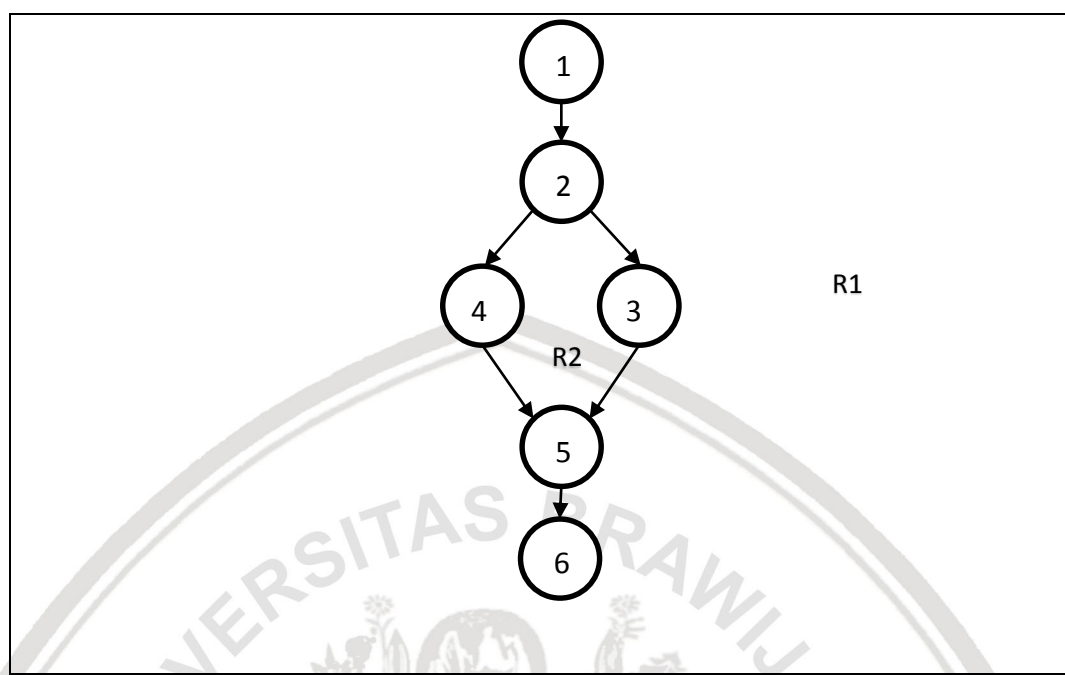
Start
View = layout fragment_native_camera
tvLokasi = data lokasi
tvKoordinat = data koordinat
opened = hasil proses dari method untuk membuka kamera
objek mGoogleApiClient
if(opened == false) (2)
    menampilkan pesan eror } (3)
    return view
else
    captureButton = data button capture
    operasi setOnClickListener untuk mengambil foto dari kamera } (4)
    mLocationRequest = request lokasi update device
    return view
endif (5)
end (6)

```



## 2. Basis Path Testing

### a. Flow Graph



### b. Cyclomatic Complexity

- $V(G) = 2$ , terdapat 2 region yaitu R1 dan R2
- $V(G) = edges - nodes + 2$   
 $= 6 - 6 + 2 = 2$
- $V(G) = predicate\ nodes + 1$   
 $= 1 + 1 = 2$

### c. Independent Path

- Jalur 1 = 1 – 2 – 4 – 5 – 6
- Jalur 2 = 1 – 2 – 3 – 4 – 6

Test case dan hasil pengujian operasi onCreateView() pada kelas CameraFragment.java akan didefinisikan pada Tabel 6.2

**Tabel 6.2 Hasil Pengujian Unit Operasi onCreateView() pada Kelas CameraFragment.java**

No	No. Jalur	Prosedur Uji	Expected Result	Result	Hasil
1	1	Input valid, opened = true	Kamera pada device akan terbuka dan Operasi akan	Kamera pada device berhasil terbuka dan Operasi akan	valid

			melanjutkan proses mengambil gambar	melanjutkan proses mengambil gambar	
2	2	Input invalid, opened = false	Operasi akan menjalankan operasi untuk membuka kamera kembali	Operasi akan menjalankan operasi untuk membuka kamera kembali	valid

### 6.1.3 Pengujian Unit Operasi getAddress() pada Kelas GeotagImage.java

#### 1. Pseudocode

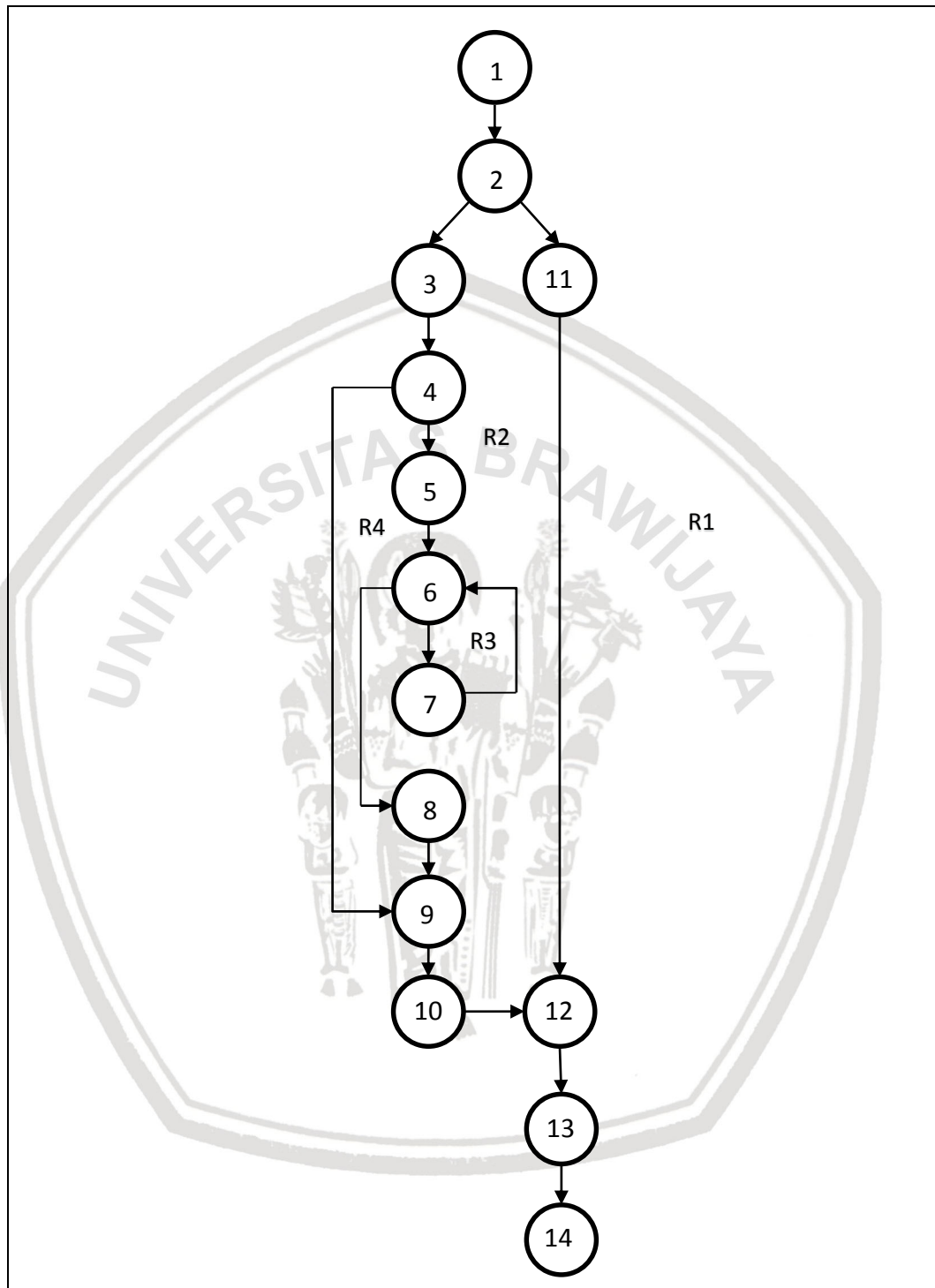
```

Start
Country = data negara
Objek Geocoder
objek strAddrBuilder
try (2)
    addressList = data lokasi dari latitude dan longitude yang ada (3)
    if (addressList != null) (4)
        data addressList pada list pertama (5)
        for(i=0 to i<list alamat maksimal) (6)
            menambah data alamat[i] (7)
        end for (8)
    end if (9)
end try (10)
catch
    printStackTrace() (11)
end catch (12)
return data alamat (13)
end (14)

```

## 2. Basis Path Testing

### a. Flow Graph



### b. Cyclomatic Complexity

- $V(G) = 4$ , terdapat 4 region = R1, R2, R3, dan R4
- $V(G) = \text{edges} - \text{nodes} + 2$

$$= 16 - 14 + 2 = 4$$

- $V(G) = \text{predicate nodes} + 1$   
 $= 3 + 1 = 4$

c. *Independent Path*

- Jalur 1 = 1 – 2 – 11 – 12 – 13 – 14
- Jalur 2 = 1 – 2 – 3 – 4 – 9 – 10 – 12 – 13 – 14
- Jalur 3 = 1 – 2 – 3 – 4 – 5 – 6 – 8 – 9 – 10 – 12 – 13 – 14
- Jalur 4 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 6 – 8 – 9 – 10 – 12 – 13 – 14

Test case serta hasil pengujian operasi getAddress() pada kelas GeotagImage.java akan didefinisikan pada Tabel 6.3

**Tabel 6.3 Hasil Pengujian Unit Operasi getAddress() pada Kelas GeotagImage.java**

No	No. Jalur	Prosedur Uji	Expected Result	Result	Hasil
1	1	Invalid input, dimana data lokasi = null	Operasi akan mengeluarkan nilai error	Operasi berhasil mengeluarkan nilai error	Valid
2	2	Invalid input, addressList = null	Operasi akan mengeluarkan nilai alamat null	Operasi berhasil mengeluarkan nilai alamat null	Valid
3	3	Valid input, addressList = 2	Operasi akan mengembalikan nilai alamat yang telah tersimpan	Operasi berhasil mengembalikan nilai alamat yang telah tersimpan	Valid
4	4	Valid input, dimana i<list alamat maksimal	Operasi akan melakukan perulangan kembali	Operasi berhasil mengeluarkan perulangan kembali	Valid

## 6.2 Pengujian Validasi

Pengujian validasi merupakan pengujian *blackbox* yang memiliki tujuan mengetahui apakah perangkat lunak yang dikembangkan telah sesuai dengan perancangan yang sebelumnya dibuat serta memenuhi semua kebutuhan yang dibutuhkan. Dari pengujian *blackbox* akan diketahui apakah fungsi-fungsi yang ada telah bekerja sesuai dengan hasil yang diharapkan.

### 6.2.1 Pengujian Validasi Melakukan Registrasi

A. Kasus uji berhasil melakukan registrasi

**Tabel 6.4 Kasus Uji Berhasil Melakukan Registrasi**

Nama Kasus Uji	Kasus uji berhasil melakukan registrasi
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Menekan tombol “Belum punya akun? Register Sekarang”</li> <li>2. Mengisi <i>field</i> Nama</li> <li>3. Mengisi <i>field</i> No Identitas</li> <li>4. Mengisi <i>field</i> Username</li> <li>5. Mengisi <i>field</i> Email</li> <li>6. Mengisi <i>field</i> Password</li> <li>7. Mengisi <i>field</i> NoHP</li> <li>8. Menekan tombol Register</li> </ol>
<b>Hasil yang diharapkan</b>	Menampilkan halaman Login
<b>Hasil</b>	Menampilkan halaman Login
<b>Status</b>	Valid

B. Kasus uji gagal melakukan registrasi dengan tidak mengisi *field username*

**Tabel 6.5 Kasus Uji Gagal Melakukan Registrasi dengan Tidak Mengisi Field Username**

Nama Kasus Uji	Kasus uji gagal melakukan registrasi dengan tidak mengisi <i>field username</i>
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Menekan tombol “Belum punya akun? Register Sekarang”</li> <li>2. Mengisi <i>field</i> Nama</li> <li>3. Mengisi <i>field</i> No Identitas</li> <li>4. Tidak mengisi <i>field</i> Username</li> <li>5. Mengisi <i>field</i> Email</li> <li>6. Mengisi <i>field</i> Password</li> <li>7. Mengisi <i>field</i> NoHP</li> <li>8. Menekan tombol Register</li> </ol>
<b>Hasil yang diharapkan</b>	Menampilkan pesan kesalahan data diri anda belum lengkap!



Hasil	Menampilkan pesan kesalahan data diri anda belum lengkap!
Status	Valid

### 6.2.2 Pengujian Validasi Melakukan *Login User*

#### A. Kasus Uji Berhasil Melakukan *Login User*

**Tabel 6.6 Kasus Uji Berhasil Melakukan *Login User***

<b>Nama Kasus Uji</b>	Kasus uji berhasil melakukan <i>login user</i>
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Mengisi <i>field Username</i> sesuai yang di daftarkan</li> <li>2. Mengisi <i>field Password</i> sesuai yang di daftarkan</li> <li>3. Menekan tombol Login</li> </ol>
<b>Hasil yang diharapkan</b>	Menampilkan halaman Home
<b>Hasil</b>	Menampilkan halaman Home
<b>Status</b>	Valid

#### B. Kasus Uji Gagal Melakukan Login User dengan Menginputkan *Password* yang Salah

**Tabel 6.7 Kasus Uji Gagal Melakukan *Login User* dengan Menginputkan *Password* yang Salah**

<b>Nama Kasus Uji</b>	Kasus uji gagal melakukan <i>login user</i> dengan menginputkan <i>password</i> yang salah
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Mengisi <i>field Username</i> sesuai yang didaftarkan</li> <li>2. Mengisi <i>field Password</i> yang tidak sesuai dengan yang di daftarkan</li> <li>3. Menekan tombol Login</li> </ol>
<b>Hasil yang diharapkan</b>	Menampilkan pesan kesalahan <i>Username</i> atau <i>Password</i> anda salah!
<b>Hasil</b>	Menampilkan pesan kesalahan <i>Username</i> atau <i>Password</i> anda salah!
<b>Status</b>	Valid

### 6.2.3 Pengujian Validasi Melihat *Home User*

#### A. Kasus Uji Berhasil Melihat *Home User*

**Tabel 6.8 Kasus Uji Berhasil Melihat *Home User***

<b>Nama Kasus Uji</b>	Kasus uji berhasil melihat <i>home user</i>
<b>Prosedur</b>	1. Menekan tab “ <i>home</i> ”
<b>Hasil yang diharapkan</b>	Menampilkan halaman <i>home</i>
<b>Hasil</b>	Menampilkan halaman <i>home</i>
<b>Status</b>	Valid

### 6.2.4 Pengujian Validasi Melaporkan Keluhan

#### A. Kasus Uji Berhasil Melaporkan Keluhan

**Tabel 6.9 Kasus Uji Berhasil Melaporkan Keluhan**

<b>Nama Kasus Uji</b>	Kasus uji berhasil melaporkan keluhan
<b>Prosedur</b>	1. Menekan tab lapor 2. Mengambil gambar yang akan dilaporkan 3. Mengisi field deskripsi 4. Memilih bagian pelaporan 5. Menekan tombol submit
<b>Hasil yang diharapkan</b>	Menampilkan halaman home dan notifikasi laporan berhasil dikirim
<b>Hasil</b>	Menampilkan halaman home dan notifikasi laporan berhasil dikirim
<b>Status</b>	Valid

#### B. Kasus Uji Gagal Melaporkan Keluhan dengan Tidak Mengisi *Field* Deskripsi

**Tabel 6.10 Kasus Uji Gagal Melaporkan Keluhan dengan Tidak Mengisi *Field* Deskripsi**

<b>Nama Kasus Uji</b>	Kasus uji gagal melaporkan keluhan dengan tidak mengisi field deskripsi
<b>Prosedur</b>	1. Menekan tab lapor 2. Mengambil gambar yang akan dilaporkan 3. Tidak mengisi field deskripsi 4. Memilih bagian pelaporan 5. Menekan tombol <i>submit</i>

<b>Hasil yang diharapkan</b>	Menampilkan pesan kesalahan “Deskripsi tidak boleh kosong!”
<b>Hasil</b>	Menampilkan pesan kesalahan “Deskripsi tidak boleh kosong!”
<b>Status</b>	Valid

## 6.2.5 Pengujian Validasi Mengambil Gambar dari Kamera

### A. Kasus Uji Berhasil Mengambil Gambar dari Kamera

**Tabel 6.11 Kasus Uji Berhasil Mengambil Gambar dari Kamera**

<b>Nama Kasus Uji</b>	Kasus uji berhasil mengambil gambar dari kamera
<b>Prosedur</b>	1. Menekan tab kamera 2. Menekan ikon <i>capture</i> untuk mengambil gambar
<b>Hasil yang diharapkan</b>	Menampilkan form pelaporan keluhan dengan gambar yang telah diambil
<b>Hasil</b>	Menampilkan form pelaporan keluhan dengan gambar yang telah diambil
<b>Status</b>	Valid

### B. Kasus Uji Gagal Mengambil Gambar dari Kamera dengan Tidak Menyalakan GPS Smartphone

**Tabel 6.12 Kasus Uji Gagal Mengambil Gambar dari Kamera dengan Tidak Menyalakan GPS Smartphone**

<b>Nama Kasus Uji</b>	Kasus uji gagal mengambil gambar dari kamera dengan tidak menyalakan GPS Smartphone
<b>Prosedur</b>	1. Menekan tab kamera 2. Menekan ikon <i>capture</i> untuk mengambil gambar
<b>Hasil yang diharapkan</b>	Menampilkan pesan kesalahan “Foto Tidak Memiliki Koordinat! Silahkan cek GPS anda atau pilih foto dari galeri”
<b>Hasil</b>	Menampilkan pesan kesalahan “Foto Tidak Memiliki Koordinat! Silahkan cek GPS anda atau pilih foto dari galeri”
<b>Status</b>	Valid

### 6.2.6 Pengujian Validasi Mengambil Gambar dari *Gallery*

#### A. Kasus Uji Berhasil Mengambil Gambar dari *Gallery*

**Tabel 6.13 Kasus Uji Berhasil Mengambil Gambar dari *Gallery***

<b>Nama Kasus Uji</b>	Kasus uji berhasil mengambil gambar dari <i>gallery</i>
<b>Prosedur</b>	1. Menekan tab <i>gallery</i> 2. Memilih gambar yang akan dilaporkan
<b>Hasil yang diharapkan</b>	Menampilkan form pelaporan keluhan dengan gambar yang telah dipilih
<b>Hasil</b>	Menampilkan form pelaporan keluhan dengan gambar yang telah dipilih
<b>Status</b>	Valid

#### B. Kasus Uji Gagal Mengambil Gambar dari *Gallery* dengan Mengambil Gambar yang Tidak Memiliki Informasi Lokasi

**Tabel 6.14 Kasus Uji Gagal Mengambil Gambar dari *Gallery* dengan Mengambil Gambar yang Tidak Memiliki Informasi Lokasi**

<b>Nama Kasus Uji</b>	Kasus uji gagal mengambil gambar dari <i>gallery</i> dengan mengambil gambar yang tidak memiliki informasi lokasi
<b>Prosedur</b>	1. Menekan tab <i>gallery</i> 2. Memilih gambar yang akan dilaporkan yang tidak memiliki informasi lokasi
<b>Hasil yang diharapkan</b>	Menampilkan pesan kesalahan “Foto Tidak Memiliki Koordinat! Silahkan cek GPS anda atau pilih foto dari galeri”
<b>Hasil</b>	Menampilkan pesan kesalahan “Foto Tidak Memiliki Koordinat! Silahkan cek GPS anda atau pilih foto dari galeri”
<b>Status</b>	Valid

### 6.2.7 Pengujian Validasi Melihat Laporan Saya

#### A. Kasus Uji Berhasil Melihat laporan Saya

**Tabel 6.15 Kasus Uji Berhasil Melihat laporan Saya**

<b>Nama Kasus Uji</b>	Kasus uji berhasil melihat laporan saya
<b>Prosedur</b>	1. Menekan tab laporan saya
<b>Hasil yang diharapkan</b>	Menampilkan daftar laporan user
<b>Hasil</b>	Menampilkan daftar laporan user
<b>Status</b>	Valid

### 6.2.8 Pengujian Validasi Melihat Detail Laporan oleh User

A. Kasus Uji Berhasil Melihat Detail Laporan oleh User

**Tabel 6.16 Kasus Uji Berhasil Melihat Detail Laporan oleh User**

<b>Nama Kasus Uji</b>	Kasus uji berhasil melihat detail laporan oleh user
<b>Prosedur</b>	1. Menekan gambar laporan yang diinginkan 2. Menekan menu “detail laporan”
<b>Hasil yang diharapkan</b>	Menampilkan detail laporan dari laporan yang dipilih user
<b>Hasil</b>	Menampilkan detail laporan dari laporan yang dipilih user
<b>Status</b>	Valid

### 6.2.9 Pengujian Validasi Merubah Laporan

A. Kasus Uji Berhasil Merubah Laporan

**Tabel 6.17 Kasus Uji Berhasil Merubah Laporan**

<b>Nama Kasus Uji</b>	Kasus uji berhasil merubah laporan
<b>Prosedur</b>	1. Menekan gambar laporan yang ingin dirubah 2. Menekan menu “ubah” 3. Merubah data laporan 4. Menekan tombol submit
<b>Hasil yang diharapkan</b>	Menampilkan halaman laporan saya
<b>Hasil</b>	Menampilkan halaman laporan saya
<b>Status</b>	Valid

B. Kasus Uji Gagal Merubah Laporan dengan Tidak Mengisi *Field* Deskripsi

**Tabel 6.18 Kasus Uji Gagal Merubah Laporan dengan Tidak Mengisi *Field* Deskripsi**

<b>Nama Kasus Uji</b>	Kasus uji gagal merubah laporan dengan tidak mengisi <i>field</i> deskripsi
<b>Prosedur</b>	1. Menekan gambar laporan yang diinginkan 2. Menekan menu “ubah” 3. Merubah data laporan dengan mengosongkan <i>field</i> deskripsi 4. Menekan tombol <i>submit</i>



Hasil yang diharapkan	Menampilkan pesan kesalahan “Deskripsi tidak boleh kosong!”
Hasil	Menampilkan pesan kesalahan “Deskripsi tidak boleh kosong!”
Status	Valid

### 6.2.10 Pengujian Validasi Menghapus Laporan User

A. Kasus Uji Berhasil Menghapus Laporan

**Tabel 6.19 Kasus Uji Berhasil Menghapus Laporan**

Nama Kasus Uji	Kasus uji berhasil menghapus laporan user
Prosedur	1. Menekan gambar laporan yang ingin dihapus 2. Menekan menu “hapus”
Hasil yang diharapkan	Menampilkan halaman laporan saya dengan laporan yang telah dihapus tidak tertampil di halaman laporan saya
Hasil	Menampilkan halaman laporan saya dengan laporan yang telah dihapus tidak tertampil di halaman laporan saya
Status	Valid

### 6.2.11 Pengujian Validasi Melakukan Logout User

A. Kasus Uji Berhasil melakukan Logout

**Tabel 6.20 Kasus Uji Berhasil melakukan Logout**

Nama Kasus Uji	Kasus uji berhasil melakukan logout
Prosedur	1. Menekan tombol logout
Hasil yang diharapkan	Menampilkan halaman login
Hasil	Menampilkan halaman login
Status	Valid

### 6.2.12 Pengujian Validasi Melakukan Login Admin

A. Kasus Uji Berhasil Melakukan Login Admin

**Tabel 6.21 Kasus Uji Berhasil Melakukan Login Admin**

Nama Kasus Uji	Kasus uji berhasil melakukan login admin
Prosedur	1. Mengisi <i>field Username</i> sesuai yang di daftarkan 2. Mengisi <i>field Password</i> sesuai yang di daftarkan

	3. Menekan tombol <i>Login</i>
Hasil yang diharapkan	Menampilkan halaman Daftar Laporan
Hasil	Menampilkan halaman Daftar Laporan
Status	Valid

B. Kasus Uji Gagal Melakukan *Login* Admin dengan Mengisi *Username* yang Salah

**Tabel 6.22 Kasus Uji Gagal Melakukan *Login* Admin dengan Mengisi *Username* yang Salah**

Nama Kasus Uji	Kasus uji gagal melakukan <i>login</i> admin dengan mengisi <i>username</i> yang salah
Prosedur	1. Mengisi <i>field Username</i> tidak sesuai yang di daftarkan 2. Mengisi <i>field Password</i> sesuai yang di daftarkan 3. Menekan tombol <i>Login</i>
Hasil yang diharapkan	Menampilkan pesan kesalahan
Hasil	Menampilkan pesan kesalahan
Status	Valid

### 6.2.13 Pengujian Validasi Melihat Daftar Laporan

A. Kasus Uji Berhasil Melihat Daftar Laporan

**Tabel 6.23 Kasus Uji Berhasil Melihat Daftar Laporan**

Nama Kasus Uji	Kasus uji berhasil melihat daftar laporan
Prosedur	1. Menekan tab daftar laporan
Hasil yang diharapkan	Menampilkan daftar laporan user
Hasil	Menampilkan daftar laporan user
Status	Valid

### 6.2.14 Pengujian Validasi Melihat Detail Laporan oleh Admin

A. Kasus Uji Berhasil Melihat Detail Laporan oleh Admin

**Tabel 6.24 Kasus Uji Berhasil Melihat Detail Laporan oleh Admin**

Nama Kasus Uji	Kasus uji berhasil melihat detail laporan oleh admin
Prosedur	1. Menekan tombol “lihat” pada laporan yang diinginkan

<b>Hasil yang diharapkan</b>	Menampilkan halaman detail laporan yang telah dipilih
<b>Hasil</b>	Menampilkan halaman detail laporan yang telah dipilih
<b>Status</b>	Valid

### 6.2.15 Pengujian Validasi Merubah Status Laporan

#### A. Kasus Uji Berhasil Merubah Status Laporan

**Tabel 6.25 Kasus Uji Berhasil Merubah Status Laporan**

<b>Nama Kasus Uji</b>	Kasus uji berhasil merubah status laporan
<b>Prosedur</b>	1. Menekan tombol “edit” 2. Memilih status laporan yang baru 3. Menekan tombol submit
<b>Hasil yang diharapkan</b>	Menampilkan halaman daftar laporan
<b>Hasil</b>	Menampilkan halaman daftar laporan
<b>Status</b>	Valid

### 6.2.16 Pengujian Validasi Menghapus Laporan oleh Admin

#### A. Kasus Uji Berhasil Menghapus Laporan oleh Admin

**Tabel 6.26 Kasus Uji Berhasil Menghapus Laporan oleh Admin**

<b>Nama Kasus Uji</b>	Kasus uji berhasil menghapus laporan oleh Admin
<b>Prosedur</b>	1. Menekan tombol “hapus” pada laporan yang ingin dihapus
<b>Hasil yang diharapkan</b>	Menampilkan halaman daftar laporan dengan laporan yang dihapus tidak tertampil di daftar laporan
<b>Hasil</b>	Menampilkan halaman daftar laporan dengan laporan yang dihapus tidak tertampil di daftar laporan
<b>Status</b>	Valid

### 6.2.17 Pengujian Validasi Melihat Daftar Pelanggan

#### A. Kasus Uji Berhasil Melihat Daftar Pelanggan

**Tabel 6.27 Kasus Uji Berhasil Melihat Daftar Pelanggan**

<b>Nama Kasus Uji</b>	Kasus uji berhasil melihat daftar pelanggan
<b>Prosedur</b>	1. Menekan tab daftar pelanggan

Hasil yang diharapkan	Menampilkan daftar pelanggan yang terdaftar
Hasil	Menampilkan daftar pelanggan yang terdaftar
Status	Valid

### 6.2.18 Pengujian Menghapus Pelanggan

#### A. Kasus Uji Berhasil Menghapus Pelanggan

**Tabel 6.28 Kasus Uji Berhasil Menghapus Pelanggan**

Nama Kasus Uji	Kasus uji berhasil menghapus pelanggan
Prosedur	1. Menekan tombol “hapus” pada pelanggan yang ingin dihapus
Hasil yang diharapkan	Menampilkan halaman daftar pelanggan dengan pelanggan yang dihapus tidak tampil di daftar pelanggan
Hasil	Menampilkan halaman daftar pelanggan dengan pelanggan yang dihapus tidak tampil di daftar pelanggan
Status	Valid

### 6.2.19 Pengujian Validasi Melakukan Logout Admin

#### A. Kasus Uji Berhasil Melakukan Logout Admin

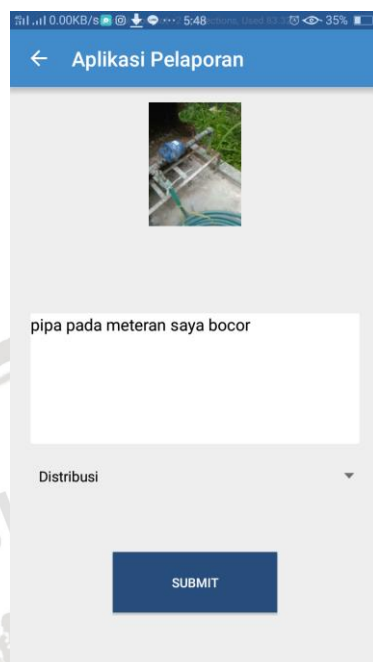
**Tabel 6.29 Kasus Uji Berhasil Melakukan Logout Admin**

Nama Kasus Uji	Kasus uji berhasil melakukan logout admin
Prosedur	1. Menekan tombol logout
Hasil yang diharapkan	Menampilkan halaman login admin
Hasil	Menampilkan halaman login admin
Status	Valid

## 6.3 Pengujian *Compatibility*

Pada pengujian *compatibility* akan dilakukan sebanyak empat kali kasus uji. Pengujian pertama yaitu menjalankan aplikasi pada android versi 5.1 Lollipop. Pengujian kedua yaitu menjalankan aplikasi pada android versi 6.0.1 Marshmallow. Pengujian ketiga yaitu menjalankan aplikasi pada android versi 7.0 Nougat. Pengujian keempat yaitu menjalankan aplikasi pada android versi 8.0 Oreo. Hasil dari pengujian *compatibility* akan dijabarkan pada Gambar 6.1 sampai Gambar 6.4 dan pada Tabel 6.30

Gambar 6.1 merupakan halaman antarmuka form laporan keluhan ketika dijalankan pada Android 5.1 Lollipop. Gambar serta tulisan dapat terlihat dengan jelas dan tampilan tidak berubah.



**Gambar 6.1 Tampilan Aplikasi pada Android Versi 5.1 Lollipop**

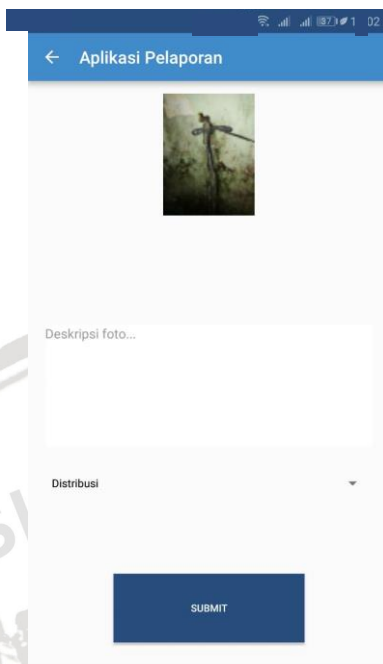
Gambar 6.2 merupakan halaman antarmuka form laporan keluhan ketika dijalankan pada Android 6.0.1 MarshMallow. Gambar serta tulisan dapat terlihat dengan jelas dan tampilan tidak berubah.



**Gambar 6.2 Tampilan Aplikasi pada Android Versi 6.0.1 MarshMallow**

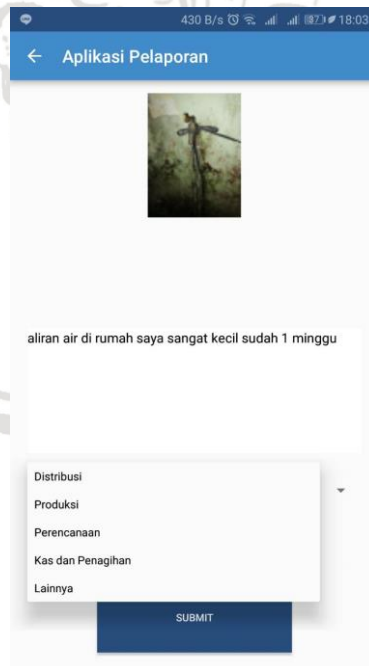


Gambar 6.3 merupakan halaman antarmuka form laporan keluhan ketika dijalankan pada Android 7.0 Nougat. Gambar serta tulisan dapat terlihat dengan jelas dan tampilan tidak berubah.



**Gambar 6.3 Tampilan Aplikasi pada Android Versi 7.0 Nougat**

Gambar 6.4 merupakan halaman antarmuka form laporan keluhan ketika dijalankan pada Android 8.0 Oreo. Gambar serta tulisan dapat terlihat dengan jelas dan tampilan tidak berubah.



**Gambar 6.4 Tampilan Aplikasi pada Android Versi 8.0 Oreo**

**Tabel 6.30 Hasil Pengujian *Compatibility***

No	Kasus Uji	Expected Result	Result	Status
1	Menjalankan aplikasi pada android versi 5.1 Lollipop	Aplikasi dapat dijalankan dengan baik dari segi fungsionalitas	Gambar, tulisan, serta menu dapat terlihat dan terbaca dengan jelas	valid
		Gambar, tulisan, serta menu dapat terlihat dan terbaca dengan jelas	Gambar, tulisan, serta menu dapat terlihat dan terbaca dengan jelas	valid
		Tata letak halaman tidak berubah	Tata letak halaman tidak berubah	valid
2	Menjalankan aplikasi pada android versi 6.0.1 MarshMallow	Aplikasi dapat dijalankan dengan baik dari segi fungsionalitas	Gambar, tulisan, serta menu dapat terlihat dan terbaca dengan jelas	valid
		Gambar, tulisan, serta menu dapat terlihat dan terbaca dengan jelas	Gambar, tulisan, serta menu dapat terlihat dan terbaca dengan jelas	valid
		Tata letak halaman tidak berubah	Tata letak halaman tidak berubah	valid
3	Menjalankan aplikasi pada android versi 7.0 Nougat	Aplikasi dapat dijalankan dengan baik dari segi fungsionalitas	Gambar, tulisan, serta menu dapat terlihat dan terbaca dengan jelas	valid
		Gambar, tulisan, serta menu dapat terlihat dan terbaca dengan jelas	Gambar, tulisan, serta menu dapat terlihat dan terbaca dengan jelas	valid
		Tata letak halaman tidak berubah	Tata letak halaman tidak berubah	valid
4	Menjalankan aplikasi pada android versi 8.0 Oreo	Aplikasi dapat dijalankan dengan baik dari segi fungsionalitas	Gambar, tulisan, serta menu dapat terlihat dan terbaca dengan jelas	valid
		Gambar, tulisan, serta menu dapat terlihat dan terbaca dengan jelas	Gambar, tulisan, serta menu dapat terlihat dan terbaca dengan jelas	valid
		Tata letak halaman tidak berubah	Tata letak halaman tidak berubah	valid

## 6.4 Analisis Hasil Pengujian

Pada tahapan ini akan dilakukan analisis terhadap pengujian yang sebelumnya telah dilakukan. Hasil dari analisis pengujian tersebut akan menghasilkan kesimpulan apakah perangkat lunak yang dibangun telah memenuhi kebutuhan yang ada dari analisis kebutuhan yang sebelumnya telah dilakukan.

#### 6.4.1 Pengujian Unit

Pada pengujian unit dilakukan pengujian pada 3 method berbeda yaitu `onClick()`, `onCreateView()`, dan `getAddress()`. Dari ketiga method tersebut akan digambarkan *basis path* yang akan menghasilkan nilai *cyclometric complexity*. Dari masing-masing method didapatkan nilai *cyclometric complexity* dan *independent path* atau jalur yang dapat dilewati pada method tersebut. Hasil yang didapat dari pengujian unit berdasarkan jalur yang ada yaitu bernilai valid yang artinya unit-unit perangkat lunak telah bekerja dengan baik.

#### 6.4.2 Pengujian Validasi

Pada pengujian validasi dilakukan uji coba pada semua fungsionalitas yang ada baik pada aplikasi *mobile* maupun *website* untuk admin. Pengujian dilakukan dengan cara menentukan kasus uji berdasarkan kemungkinan inputan yang dapat dimasukkan oleh aktor. Kemudian menjalankan kasus uji tersebut pada aplikasi yang telah dibangun. Hasil dari pengujian validasi yang dilakukan pada 11 fungsional aplikasi *mobile* pelaporan keluhan dan 8 fungsional website admin yaitu setiap kasus uji menampilkan keluaran yang sesuai dengan yang diharapkan pada saat analisis kebutuhan sehingga menghasilkan status valid. Hal ini menunjukkan bahwa perangkat lunak yang dibangun sudah memenuhi seluruh kebutuhan yang telah didefinisikan pada saat analisis kebutuhan.

#### 6.4.3 Pengujian *Compatibility*

Pada pengujian *compatibility* ini dilakukan pengujian dengan cara menjalankan aplikasi mobile pelaporan keluhan pada 4 android dengan versi OS yang berbeda. Hasil pengujian menyatakan bahwa aplikasi dapat dijalankan dengan baik pada Android versi 5.1 Lollipop, Android 6.0.1 Marshmallow, Android 7.0 Nougat, dan Android 8.0 Oreo. Pada saat dijalankan, seluruh fungsionalitas dapat berjalan dengan baik dan tampilan baik font, gambar, maupun menu yang ada dapat terlihat dengan jelas.

## BAB 7 KESIMPULAN DAN SARAN

### 7.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan dalam beberapa tahap ini yang dimulai dari analisis kebutuhan, perancangan, implementasi, dan pengujian sistem, hasil yang didapatkan dapat ditarik kesimpulan sebagai berikut:

1. Pada proses analisis kebutuhan sistem yang dilakukan berdasarkan hasil observasi, didapatkan hasil berupa 19 kebutuhan fungsional yang terdiri dari 11 kebutuhan fungsional aplikasi *mobile* dan 8 kebutuhan fungsional *website* admin serta 1 kebutuhan non fungsional yang dibutuhkan pelanggan untuk melaporkan keluhan kepada PDAM Tirta Tuah Benua dan karyawan untuk memberikan umpan balik kepada pelanggan. Kemudian didefinisikan pemodelan kebutuhan sistem yang berfungsi sebagai acuan dalam melakukan perancangan sistem yaitu berupa *use case* diagram dan *use case* skenario. Pada proses perancangan dilakukan berdasarkan analisis kebutuhan yang telah dilakukan. Pada tahap ini didapatkan hasil berupa perancangan arsitektur sistem, perancangan basis data yang di representasikan dengan ERD (*Entity Relationship Diagram*), perancangan *sequence* diagram, perancangan *class* diagram, perancangan algoritme yang akan digunakan pada saat implementasi dan yang terakhir perancangan antarmuka yang menggambarkan antarmuka dari aplikasi yang akan dibangun.
2. Pada proses implementasi, hasil yang didapat yaitu implementasi basis data dalam bentuk PDM (*Physical Data Model*), implementasi kode program dan yang terakhir implementasi antarmuka. Pada implementasi kode program digunakan *Object Oriented Programming* dengan bahasa pemrograman Java menggunakan *software* Android Studio untuk aplikasi *mobile* serta bahasa pemrograman PHP untuk *website* admin dengan menggunakan *framework* *Code Igniter*.
3. Pada proses pengujian validasi yang telah dilakukan pada 19 fungsional didapatkan hasil bahwa semua fungsional bernilai valid yang berarti telah sesuai dengan analisis kebutuhan yang telah dilakukan sebelumnya. Pada proses pengujian *compatibility* didapatkan hasil bahwa aplikasi *mobile* dapat digunakan pada 4 versi OS Android yang berbeda yaitu Android 5.1 Lollipop, Android 6.0.1 MarshMallow, Android 7.0 Nougat, dan Android 8.0 Oreo.

### 7.2 Saran

Berdasarkan hasil yang telah didapatkan, saran untuk penelitian ini bagi pengembang selanjutnya antara lain:

1. Informasi yang diterima aplikasi bisa lebih di spesifikkan
2. Ditambahkan fitur *direction* untuk mempermudah karyawan PDAM menuju lokasi pelanggan.

3. Dapat dilakukan pengembangan aplikasi pelaporan keluhan pada berbagai platform lain seperti iOS, *Windows Phone*, sehingga tidak hanya pengguna Android yang bisa menggunakan aplikasi ini.





## DAFTAR PUSTAKA

- Albharkah, G. C., 2016. *Rancang Bangun Aplikasi Mobile Social Geotagging Informasi Barang Hilang Pada Platform Android*. Malang: Informatika.
- Anwar, S. N., Nugroho, I. & Lestariningsih, E., 2015. Perancangan dan Implementasi Aplikasi Mobile Semarang Guidance Pada Android. *Jurnal Teknologi Informasi DINAMIK*, Volume 20, pp. 148-158.
- Ariyansyah, E., 2016. *Rancang Bangun Aplikasi Mobile Social Geotagging Keragaman Budaya Indonesia Pada Platform Android*. Malang: Universitas Brawijaya.
- Bhat, A. & Quadri, S., 2015. *Equivalence Class Partitioning and Boundary Value analysis*. s.l., Sustainable Global Development (INDIACom).
- Bintarawan, W. A., 2013. *Pengembangan Sistem Pengaduan Pelanggan pada Perusahaan Daerah Air Minum Kota Salatiga*. Salatiga: Universitas Kristen Satya Wacana.
- Fauzia, S. Q., 2017. *Rancang Bangun Sistem Pelaporan Keluhan Masyarakat Dengan Fitur Geotagging Berbasis Android*. Malang: Universitas Brawijaya.
- Firly, N., 2018. *Create Your Own Android Application*. Jakarta: PT Elex Media Komputindo.
- H.N, I. A., Nugroho, P. I. & Ferdiana, R., 2015. Pengujian Usability Website Menggunakan System Usability Scale. *IPTEKKOM: Jurnal Penelitian Teknologi Informasi dan Komunikasi*, Volume 17.
- Huda, A. A., 2012. *24 Jam!! Pintar Pemrograman Android*. Yogyakarta: Andi Offset.
- Istiyanto, J. E., 2013. *Pemrograman Smart Phone Menggunakan SDK Android dan Hacking Android*. Yogyakarta: Graha Ilmu.
- Kumar, M., Kant, K., Varyani, B. & Kale, M., 2015. *Analysis of Optimization Requirement of Mobile Application Testing Procedure*. s.l., ICSEA 2015: The Tenth International Conference on Software Engineering Advances.
- Minarni & Yusdi, Y. F., 2015. Sistem Informasi Geografis Pariwisata Kota Padang Menggunakan Application Programming Interface (API) Google Maps Berbasis Web. *Teknolf*, Volume 3, pp. 31-37.
- Mustaqbal, M. S., Firdaus, F. R. & Rahmadi, H., 2015. Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi kasus: Aplikasi Prediksi Kelulusan SNMPTN). *ISSN*, pp. 2407-3911.
- Mustaqbal, M. S., Firdaus, R. F. & Rahmadi, H., 2015. Pengujian Palikasi Menggunakan Black Box Testing Boundary Value Analysis. *Jurnal Ilmiah Teknologi Informasi Terapan*, Volume 1.
- MZ, M. H., 2017. Aplikasi Rekomendasi Spot Area Wisata Berbasis ANDroid dengan Teknik Geotag. *Jurnal INFORM*, Volume 2.

Nandipati, A., 2011. Assesment of Metadata Associated. *Institute for Geoinformatics University of Muenster*.

Nielsen, J., 2012. *Usability 101: Introduction to Usability*. s.l.:s.n.

Pressman, R. S. & Maxim, B. R., 2015. *Software Engineering: A Practitioner's Approach*. Eighth Edition ed. New York: McGraw-Hill.

Simarmata, J., 2010. *Rekayasa Perangkat Lunak*. Yogyakarta: Andi.

Sommerville, I., 2011. *Software Engineering: Ninth Edition*, [e-book] Boston: Pearson Education.

S, R. A. & Shalahuddin, M., 2016. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.

Tirta Tuah Benua, P., 2017. *PDAM Tirta Tuah Benua Kutai Timur*. [Online] Tersedia di: <http://pdam-kutaitimur.com/statis-2/visi-dan-misi.html> [diakses pada 2 Desember 2017].

Wijaya, W., Tolle, H. & Kharisma, A. P., 2018. Rancang Bangun Aplikasi Geotagging Social Report Bencana Banjir. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2, pp. 2817-2824.

Wiriyo, F. H., 2016. *Pengembangan Aplikasi Mobile Pelaporan Keluhan Pelanggan Listrik Menggunakan Fitur Location Based Services Berbasis Android*. Malang: Universitas Brawijaya.

Y., 2013. *Cara Mudah & Murah Membangun & Mengelola Website*. Yogyakarta: Graha Ilmu.